

**This Page Is Inserted by IFW Operations
and is not a part of the Official Record**

BEST AVAILABLE IMAGES

**Defective images within this document are accurate representation of
The original documents submitted by the applicant.**

Defects in the images may include (but are not limited to):

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORLED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PAT-NO: JP403184109A

DOCUMENT-IDENTIFIER: JP 03184109 A

TITLE: TARGET DESIGNATION RESET METHOD FOR DATA PROCESSOR

PUBN-DATE: August 12, 1991

INVENTOR-INFORMATION:

NAME

BRUCKERT, WILLIAM

KOVALCIN, DAVID

BISSETT, THOMAS D

MUNZER, JOHN

NORCROSS, MITCHELL

ASSIGNEE-INFORMATION:

NAME

COUNTRY

DIGITAL EQUIP CORP <DEC>

N/A

APPL-NO: JP02203805

APPL-DATE: July 31, 1990

INT-CL (IPC): G06F001/24, G06F011/20, G06F015/16

ABSTRACT:

PURPOSE: To prevent the reset state occurring in one zone from being automatically propagated to another zone by generating a reset signal only for each of zones in a multiple zone processing system.

CONSTITUTION: In a fault tolerant computer system 10, zones 11 and 11' are simultaneously operating. Zones 11 and 11' have duplicate processing systems 20 and 20' respectively. Systems 20 and 20' have plural modules connected to each other. When a data processor having this constitution will be reset, the transaction sent on a data path at present is stored. Next, the state of a system to which reset is indicated is detected. When the reset state is detected, the reset signal is transmitted to a selected component through the data path to reset the selected component. Thereafter, the stored current transaction is sent again through the data path.

COPYRIGHT: (C)1991,JPO

⑫ 公開特許公報(A) 平3-184109

⑤ Int. Cl.⁵

識別記号

庁内整理番号

⑬ 公開 平成3年(1991)8月12日

G 06 F 1/24
11/20

3 1 0 Z

9072-5B
7459-5B

G 06 F 1/00 3 5 0 B※

審査請求 未請求 請求項の数 17 (全52頁)

⑭ 発明の名称 データ処理装置における目標指定リセット法

⑯ 特 願 平2-203805

⑰ 出 願 平2(1990)7月31日

優先権主張 ⑱ 1989年8月1日 ⑲ 米国(US) ⑳ 388087

㉑ 発 明 者 ウィリアム ブルツカ アメリカ合衆国 マサチューセッツ州 01532 ノースボ
ート ロ マツシュビー サークル 13㉒ 発 明 者 デイヴィッド コヴァ アメリカ合衆国 マサチューセッツ州 01519 グラフト
ルシン シェリル ドライブ 8㉓ 出 願 人 デジタル イクイブ アメリカ合衆国 マサチューセッツ州 01754 メイナ
メント コーポレーシ ド メイン ストリート 146番
ョン㉔ 代 理 人 弁理士 中 村 稔 外7名
最終頁に続く

明細書の浄書(内容に変更なし)

明 細 書

1. 発明の名称 データ処理装置における目標指
定リセット法

2. 特許請求の範囲

1. データ通路を介し複数の構成要素と接続する
中央処理装置を有し、これら構成要素がリセッ
ト可能な要素を含み、該中央処理装置がデータ
通路に沿って送られる一連のトランザクション
を起す命令のシーケンスを実行するデータ処理
装置におき命令実行のシーケンスを変更するこ
となしにデータ処理装置をリセットする方法に
おいて、データ通路上を現在送られているトランザク
ションを格納する段階と、リセットが指示されたデータ処理システムの
状態を検出する段階と、リセットの状態が探知された場合に、複数の
構成要素のうちの選択されたものにデータ通路
に沿ってリセット信号を送信して、このリセッ
ト信号によってこれらの選択された構成要素に

その部分をリセットさせる段階と、

格納された現行のトランザクションをデータ
通路に沿って再送する段階と、を含む命令実行のシーケンスを変更すること
なしにデータ処理装置をリセットする方法。2. 前記選択された構成要素のリセット可能な要
素がそれぞれその状態を識別するインディケー
タを有し、前記リセット信号を送信する段階が前記構成
要素の状態インディケータをリセットするため
の従属段階を含むことを特徴とする請求項1記
載の方法。3. 前記選択された構成要素のリセット可能な要
素のそれぞれが前記一連のトランザクションの
途中においてデータ通路に沿って送信されるデ
ータを格納するための少なくとも一つの記憶レ
ジスタを有し、前記リセット信号を送信する段階が前記選択
された構成要素の記憶レジスタをリセットする
従属段階を含むことを特徴とする請求項1記載

の方法。

4. 前記選択された構成要素のリセット可能な要素のそれぞれがエラー情報を含む少なくとも一つのエラー回路を有し、

前記リセット信号を送信する段階が前記選択された構成要素のエラー回路をリセットする従属段階を含むことを特徴とする請求項1記載の方法。

5. 前記リセットを指示された状態を感知する段階がエラー状態を検出するための従属段階を含むことを特徴とする請求項1記載の方法。

6. 前記リセットを指示された状態を検出する段階がリセット要求状態を感知する従属段階を含むことを特徴とする請求項1記載の方法。

7. データ通路を介し複数の構成要素と接続する中央処理装置を有し、これら構成要素がリセット可能な要素を含み、該中央処理装置がデータ通路に沿って送られる一連のトランザクションを起す命令のシーケンスを実行するデータ処理装置における自動的にデータ処理システムをリ

セットする方法において、

データ通路上を現在送られているトランザクションを格納する段階と、

リセットが指示されたデータ処理システムの状態を検出する段階と、

指示されたりセットがクリティカルまたはノンクリティカルなりセット状態のいずれであるかを判定する段階と、

指示された状態がクリティカルなりセット状態である場合に前記複数の構成要素にハードリセット信号を発行し、このハードリセット信号の発行がリセット可能な要素の総てをリセットさせ且つデータ処理システムを所定の状態に入らせ、したがって前記データ処理システムによる命令実行の通常のシーケンスを壊すようにハードリセット信号を発行する段階と、

指示された状態がノンクリティカルなりセット状態である場合に前記複数の構成要素のうちの選択されたものに対してソフトリセット信号を発行し、これら選択された構成要素によるソ

フトリセット信号の受信がデータ処理システムの命令実行の通常のシーケンスの中断をさせないようにソフトリセット信号を発行する段階と、

ソフトリセット信号状態の発行の後に格納された現行のトランザクションをデータ通路に沿って再送する段階と

を含むことを特徴とするデータ処理システムを自動的にリセットする方法。

8. 前記指示されたりセットがクリティカル或いはノンクリティカルリセット状態のいずれであるかを判定する段階が、

電力投入信号が受信されてデータ処理システムに対して電力が最近印加されたことを示す場合に指示されたりセットがクリティカルなりセット状態であると判定する従属段階を含むことを特徴とする請求項7記載の方法。

9. 前記指示されたりセットがクリティカル或いはノンクリティカルリセット状態のいずれであるかを判定する段階が

前記データ処理システムから1構成要素を除

去するように要求を受信した場合に指示されたりセットをクリティカルリセット状態と判定する従属段階を含むことを特徴とする請求項7記載の方法。

10. 前記データ処理システムが互いに同期に作動するように設計された二重処理システムを含み、

前記指示されたりセットがクリティカル或いはノンクリティカルリセット状態のいずれであるかを判定する段階が

前記二重処理システムを同期にするように要求を受けた場合に、指示されたりセットがクリティカルなりセット状態であると判定する従属段階を含むことを特徴とする請求項7記載の方法。

11. 二つのデータ処理ゾーンを有し、それぞれのゾーンがデータ通路を介して複数の構成要素に接続する中央処理装置を含み、これらの構成要素がリセット可能な要素を含み、前記中央処理装置のそれぞれが、データ通路に沿って送られる一連のトランザクションを起す命令のシーケ

ンスを実行するデータ処理システムにおけるデータ処理システムを自動的にリセットする方法において、

データ通路上を現在送られているトランザクションを格納する段階と、

リセットが指示されたデータ処理システムの状態を検出する段階と、

指示されたりセットがクリティカルまたはノンクリティカルなりセット状態のいずれであるかを判定する段階と、

指示された状態がクリティカルなりセット状態である場合に前記ゾーンの両方の複数の構成要素に対してハードリセット信号を発行し、このハードリセット信号の発行がリセット可能な要素の総てをリセットさせ且つデータ処理システムを所定の状態に入らせ、したがって前記データ処理システムによる命令実行の通常のシーケンスを壊し、このハードリセット信号の発行が前記ゾーンのそれぞれの構成要素に対してほぼ同時に生じるようにハードリセット信号を発

行する段階と、

指示された状態がノンクリティカルなりセット状態である場合に前記複数の構成要素のうちの選択されたものに対してリセット通路に沿ってソフトリセット信号を発行し、ソフトリセット信号が前記ゾーンの両方における前記選択された構成要素にほぼ同時に到着し、これら選択された構成要素によるソフトリセット信号の受信がデータ処理システムの命令実行の通常のシーケンスの中断をさせないようにソフトリセット信号を発行する段階と、

ソフトリセット信号状態の発行の後に、格納された現行のトランザクションを各データ通路に沿って再送する段階と、

を含むことを特徴とするデータ処理システムを自動的にリセットする方法。

12. 前記指示されたりセットがクリティカルまたはノンクリティカルなりセット状態のいずれであるかを判定する段階が、

電力投入信号が受信されてデータ処理システ

ムに対して電力が最近印加されたことを示す場合に指示されたりセットがクリティカルなりセット状態であると判定する従属段階を含むことを特徴とする請求項11記載の方法。

13. 前記指示されたりセットがクリティカルまたはノンクリティカルなりセット状態のいずれであるかを判定する段階が、

前記データ処理システムから1構成要素を除去するように要求を受信した場合、指示されたりセットをクリティカルなりセット状態と判定する従属段階を含むことを特徴とする請求項11記載の方法。

14. 前記指示されたりセットがクリティカルまたはノンクリティカルなりセット状態のいずれであるかを判定する段階が、

前記各ゾーンを同期にするように要求を受信した場合には指示されたりセットがクリティカルなりセット状態であると判定する従属段階を含むことを特徴とする請求項11記載の方法。

15. ソフトリセット信号を発行する段階が各ゾー

ンについてソフトリセット信号を生成して各ゾーンで生成されたソフトリセット信号を同じゾーンの選択された構成要素に送る従属段階を含むことを特徴とする請求項11記載の方法。

16. リセットが指示されたデータ処理システムの状態を感知する段階が前記各ゾーンの一つにおいて感知をなす従属段階を含み、

指示されたりセットがクリティカルまたはノンクリティカルなりセット状態のいずれであるかを判定する段階が状態を感知したのと同じゾーンにおいてそのような判定をなす従属段階を含み、

ソフトリセット信号を発行する段階がソフトリセット開始信号を各ゾーンのうちリセット状態を検出したゾーンから各ゾーンのうちのもう一方のゾーンに送る従属段階を含むことを特徴とする請求項15記載の方法。

17. それぞれが複数の構成要素を含み、ほぼ同時に同一の一連の動作を実行する二つのデータ処理システムを有するコンピュータシステムにお

けるリセットをデータ処理システム全体に伝播させる方法において、コンピュータシステムによって実行される

リセットが指示されたコンピュータシステムの状態を検出する段階と、

前記状態に対応してデータ処理システムのそれぞれによってリセット信号を独立に生成する段階と、

各データ処理システムによって生成されたリセット信号を、対応するリセット信号を生成したデータ処理システムの要素に対してのみ送信する段階とを含むことを特徴とするリセットをデータ処理システム全体に伝播する方法。

る。

従来の機器におけるリセットの別の問題はそれを局所化できないことにある。すなわち一部分だけをリセットする必要がある場合にデータ処理システム全体をリセットすることになる。このことは、多重プロセッサを例えば故障許容業務のため等に用いるシステムにおいて特に問題となる。このようなシステムにおいては各プロセッサの一つにおけるエラーが他のプロセッサに伝播し、システム全体を止めてしまうことが有り得る。

もし発信側のプロセッサがリセット信号を生成するに際して誤りを犯すと、この影響によって実行中に不必要な停止を引き起すことになる。

したがって、リセットがリセットを生じた状態と一致するシステムを設計すれば有効である。

また、このようなシステムが効果の異なる数種類のリセットを備えていれば効果的である。

さらに、多重プロセッサデータ処理システムにおいて各プロセッサの一つでのリセットが自動的に他のプロセッサに伝播しなければ効果的であ

3. 発明の詳細な説明

(産業上の利用分野)

本発明は、データ処理装置のリセットに関連し、特にデータ処理装置における異なった種類のリセットの管理の分野に関連する。

(従来の技術)

データ処理システムにおいては、電源を入れる際或いは特定の種類のエラーが発生した場合等に、特定の状況下におけるリセットの機能が必要とされる。リセットなしには、データ処理システムを初期化ルーチン或いはエラー回復ルーチンを開始する所定の状態に設定する方法はない。

(発明が解決しようとする課題)

リセットについての問題点は、それが広範な影響を及ぼすことにある。一般に、リセットは、命令実行の通常のフローを壊し、データや情報の損失を招くことがある。場合によっては、より深刻な問題を防ぐためにこのような非常手段が必要となるが、リセットの影響がリセットを起こさせた状態よりもひどいものであることがしばしばあ

る。

この発明のその他の効果は、以下の説明によってその一部が明示され、その他はその説明から明らかとなるか或いはこの発明を実施することによって知ることができる。これらの効果は、添付の特許請求の範囲において特に示す方法および装置によって理解され得るものである。

(発明を解決するための手段)

本発明は、命令の通常の実行をなすハードリセットと命令操作に対して一般に透明なソフトリセットとを区別することによって、従来技術における問題点を解決して上述した目的を達成する。加えて、リセットはシステム全体に及ぶものおよび局部的なものの両方で有り得る。そして、多重ゾーン処理システムにおける各ゾーンはそれのためのリセットを生成し、1つのゾーンで起こされたリセットが他のゾーンに自動的に伝播されないようにしている。

この発明の目的にしたがって実施され本明細書において詳細に説明される命令実行のシーケンス

を変更することなしにデータ処理装置をリセットする方法は、データ処理システムによって実行される幾つかの段階によって構成されている。このデータ処理システムはデータ通路を介して複数の構成要素に接続する中央処理装置を有している。これらの構成要素は、リセット可能な要素を含み、中央処理装置は、データ通路に沿って送られる一連のトランザクションを起こす命令シーケンスを実行する。

これらの段階には、データ通路上を現在送られているトランザクションを格納する段階と、リセットが指示されているデータ処理システムの状態を感知する段階と、リセットの状態が感知された場合に、複数の構成要素のうちの選択されたものにデータ通路に沿ってリセット信号を送信してこのリセット信号によってこれらの選択された構成要素にそれらの要素の部分をリセットさせる段階と、データ通路に沿って格納した現行のトランザクションを再送する段階とを含む。

とができる。

第2図は、故障許容コンピュータ・システム10の物理的ハードウェアを示し、システムが重複して設けられていることを図によって示す。各ゾーン11と11'は、別のキャビネット11と12'にそれぞれ内蔵されている。キャビネット12は、バッテリー13、電源調整装置14、冷却ファン16およびAC入力17を有する。キャビネット12'はキャビネット12の構成要素13、14、16および17に対応する別の構成要素を有する。

以下で詳細に説明するように、処理システム20および20'は背面板によって相互に接続された幾つかのモジュールを有する。もし1つのモジュールに故障またはエラーがあれば、このモジュールは、コンピュータ・システム10を動作不能にすることなく、取り外して取り替えることができる。これは、処理システム20と20'が物理的に分離され、モジュールがプラグによって挿入されている別の背面板を有し、相互に独立して

(実施例)

本発明の好適な実施例を詳細に参照するが、この実施例の具体例は添付図に示されている。

A. システムの説明

第1図は本発明による故障許容コンピュータ・システム10のブロック図である。この故障許容コンピュータ・システム10はゾーンと呼ぶ重複システムを有している。通常のモードの場合、2つのゾーン11と11'が同時に動作している。この重複によって、1つのポイントで故障が発生することがなく、ゾーン11または11'の1つにエラーまたは故障が発生しても、これによってコンピュータ・システム10が動作不能にならないことが保証される。さらに、こうした故障は、これを発生させた装置または構成要素を動作不能にするまたは無視することによって取り除くことができる。第1図に示すゾーン11と11'は、それぞれ重複処理システム20と20'を有している。しかし、これらが重複して設けられていることによって、処理システム以上のことを行うこ

動作することができるためである。従って、これらのモジュールは、一方の処理システムが動作を継続している間に、他方の処理システムの背面板から取り外しまたはその背面板にプラグによって挿入することができる。

好適な実施例の場合、重複処理システム20および20'は同一のものであり、同一のモジュールを内蔵している。したがって、処理システム20'は同じ動作をすると理解して、処理システム20のみを完全に説明する。

処理システム20は第3図および第4図に詳細に示すCPUモジュール30を有している。CPUモジュール30は、以下で詳細に説明するクロスリンク経路25によって処理システム20'のCPUモジュール30'と相互に接続されている。クロスリンク経路25によって、処理システム20と20'との間にデータ転送経路が設けられ、処理システム20と20'が同期して動作することを保証するためにタイミング信号が搬送される。

処理システム20はまたI/Oモジュール100、

110、および120を有する。I/Oモジュール100、110、120、100'、110'および120'は独立した装置である。第1図、第4図および第17図はI/Oモジュール100を詳細に示す。複数のI/Oモジュールを図示するが、これらの重複したモジュールはこのシステムによって要求されるものではない。しかし、このような重複がなければ、ある程度の補償許容度が失われる。

I/Oモジュール100、110、120の各々は、デュアル・レール・モジュール相互接続部130および132によってCPUモジュール30に接続される。モジュール相互接続部130と132はI/O相互接続部として機能し、背面板を介して処理システム20に接続されている。この用途に使用するため、CPU40、メモリ制御装置70、クロスリック90およびモジュール相互接続部130を有するデータ経路が一方のレールと考えられ、CPU50、メモリ制御装置75、クロスリンク95、およびモジュール相互

接続部132を有するデータ経路が他方のレールと考えられる。動作が正しく行われている間は、両方のレールのデータは同じである。

B. 故障許容システムの原理

故障許容コンピュータ・システム10では、1つのポイントで故障の発生することがないが、その理由は、各構成要素が重複して設けられているためである。処理システム20と20'は、それぞれ故障停止処理システムであり、このことは、これらのシステムがサブシステム内の故障またはエラーを検出し、これらの故障またはエラーが他のサブシステムに制御されない状態で広がることを防止することができる。しかし、これらの処理システムでは、各処理システム内の構成要素が重複して設けられていないため、1つの点で故障が発生する。

2つの故障停止処理システム20と20'は、所定の方法で動作するある種の構成要素によって相互に接続され、フェール・セーフ・システムを形成する。故障許容コンピュータ・システム10

として具体化されているフェール・セーフ・システムの場合、たとえ故障停止処理システム20および20'の一方が故障しても、コンピュータ・システムは全体として処理を継続することができる。

2つの故障停止処理システム20と20'はロックステップ同期で動作すると考えられるが、その理由は、CPU40、50、40'および50'がこのような同期で動作するからである。この場合、3つの重要な例外が存在する。第1の例外は、ブートストラップ法によって両方の処理装置を同期させる初期化の時に発生する。第2の例外は、処理システム20と20'が2つの異なる作業負荷で独立して（非同期の状態）で動作する場合に発生する。第3の例外は、ある種のエラーが処理システム20と20'に発生する場合に起こる。この最後の例外の場合、これらの処理システム内の一方のCPUとメモリ素子を動作不能し、これによって同期動作を終了する。

システムがロックステップI/Oで動作してい

る場合、いずれの1つの時間にも、1つのI/O装置のみしかアクセスすることができない。しかし、4つのCPU40、50、40'および50'は全て実質的に同じ時間に同じデータをこのI/O装置から受け取る。以下の議論では、これらの処理システムのロックステップ同期とは、1つのI/Oモジュールのみがアクセスされていることを意味すると理解できる。

重複して設けられた処理システム20および20'の同期は、各システムを決定性を有する機械として取り扱うことによって実行され、この場合、これらのシステムは、同じ入力を受けて同じ既知の状態からスタートし、常に同じ機械状態に入り、エラーのない場合には、同じ結果を発生する。処理システム20と20'は同じ構成を有し、同じ入力を受取、従って、同じ状態を通過する。従って、両方の処理装置が同期して動作する限り、これらは同じ結果を発生すると共に同じ状態に入る。もしこれらの処理システムが同じ状態でなく、または異なる結果を発生すれば、これらの処理

システム20と20'の一方が故障していると考えられる。そこで修正動作を行うためには、故障しているモジュールを動作不能にする等して故障の原因を取り除かなければならない。

エラーの検出は、一般的に別の処理時間または論理の形でのオーバーヘッドを含む。このようなオーバーヘッドを最小にするため、システムは故障許容動作と調和しながら、エラー・チェックをできるだけ少ない回数行わなければならない。少なくとも、エラーのチェックはデータがCPU30と30'から出力される前に行われなければならない。そうでなければ、内部処理のエラーによって、原子炉のような外部システムに正しくない動作が発生するが、これは故障許容システムの設計によって防止しようとしている状態である。

これ以外にエラーのチェックを行う理由が存在する。例えば、故障またはエラーを除去するためには、記憶または使用する前に、CPUモジュール30および30'の受け取ったデータをチェックすることが望ましい。そうでなければ、記憶さ

れているエラーのあるデータが後がアクセスされ、その結果、別のエラーが発生すると、特にこのエラーのあるデータが一定期間記憶された場合には、これらのエラーの最初の原因を見出すことが困難または不可能になる。時間が経過することとこれらのエラーのあるデータがその後処理されることによって、エラーの原因を追跡することができなくなる可能性がある。

検出される前にエラーが記憶されていた時間を指す「エラーの潜伏時間」によって、同様に後で問題が発生する可能性がある。例えば、コンピュータシステムが以前に発生したエラーによって小さくなったキャパシティで既に動作している場合に、滅多に使わないルーチンによって潜在するエラーの見付かる場合がある。コンピュータのキャパシティが減少している場合、潜在するエラーによってシステムが破壊される場合がある。

更に、処理システム20および20'がデュアル・レール・システムになっている場合、データをメモリのような共有の資源である1つのレール

・システムに転送するのに先立って、エラーをチェックすることが望ましい。この理由は、このような転送を行った後には最早2つの独立するデータのソースが存在しないためであり、もしシングル・レール・システムで後になって何らかのエラーが検出された場合、このエラーを追跡することは、不可能でないにしても困難になる。

エラー処理の好適な方法が、これと同じ日に出版された弁理士ドケットNo. PD 89-289 / DEC-344の発明の名称「ソフトウェアによるエラーの処理」という出願で説明され、これは参考としてここに引用されている。

C. モジュールの説明

1. CPUモジュール

第1図に示すCPUモジュール30の構成要素を第3図および第4図により詳細に示す。第3図はCPUモジュールのブロック図であり、第4図はCPUモジュール30およびI/Oモジュール100並びにこれらの相互接続部のブロック図である。CPUモジュール30および30'の動作

およびこれらに含まれる構成要素は一般的に同じであるため、CPU30のみを説明する。

CPUモジュールは、デュアルCPU40と50を内蔵する。CPU40と50は当業者に周知の標準的な中央処理装置である。好適な実施例の場合、CPU40と50は本出願の譲受人であるディジタル・エキップメント会社によって製造されたVAXマイクロプロセッサである。

CPU40と50に関連するのはそれぞれキャッシュ・メモリ42と52であり、これらはCPUに対して十分なメモリのサイズを有する標準のキャッシュRAMである。好適な実施例の場合、キャッシュRAMは4Kx64ビットである。しかし、本発明がキャッシュRAMを有する必要はない。

2. メモリ・モジュール

CPU40と50は、最高4つのメモリ・モジュール60を共有できることが望ましい。第5図はCPUモジュール30に接続して示した1つのメモリ・モジュール60のブロック図である。

メモリ転送サイクル、ステータス・レジスタ転送サイクルおよびEEPROM転送サイクルの期間中、各メモリ・モジュール60は双方向データ・バス85を介してプライマリ・メモリ制御装置70にデータを転送すると共にこれからデータの転送を受ける。各メモリモジュール60は、またそれぞれバス80および82を介してメモリ制御装置70と75からアドレス信号、制御信号、タイミング信号およびECC信号を受け取る。バス80および82のアドレス信号は、ボード信号、バンク信号、および行アドレス信号と列アドレス信号を含み、これらによってデータ転送に含まれるメモリ・ボード・アドレス、バンク・アドレス、および行および列アドレスが識別される。

第5図に示すように、各メモリ・モジュール60はメモリ・アレイ600を有する。各メモリ・アレイ600はDRAMが8バンクのメモリに組織されている標準RAMである。好適な実施例の場合、高速ページ・モード型のDRAMが使用される。

れた場合、故障の原因を判定するため、記憶されているデータがEEPROM640から取り出される。EEPROM640は、ドライバ630からの行アドレス線を介して、制御ロジック610からのEEPROM制御信号によってアドレスされる。EEPROM640は、32ビットの内部メモリ・データ・バス645に対して8ビットのデータを転送し、ここからこのデータを受け取る。

制御ロジック610は、メモリ・モジュール60の素子に対してアドレス信号を転送すると共に内部タイミングと制御信号を発生する。第6図に詳細に示すように、制御ロジック612はプライマリ/ミラー指示回路612を有する。

プライマリ/ミラー指示回路612は、バス80と82でメモリ制御装置70と75から2組のメモリ・ボード・アドレス・バンク・アドレス、行および列アドレス、サイクル・タイプ・サイクル・タイミング信号を受け取り、またバス80と82でメモリ制御装置に対して2組のECC信号を転送すると共にここからこれを受け取る。指示

メモリ・モジュール60には、また制御ロジック610、データ・トランシーバ/レジスタ620、メモリ・ドライバ630、およびEEPROM640が含まれる。データ・トランシーバ/レジスタ620によってメモリ・アレイ600とデータ・バス85の双方向データ線との間でデータを転送するためのデータ・バスとデータ・インターフェースが設けられる。メモリ・ドライバ630は、制御ロジック610からメモリ・アレイ600の各バンクに対して行および列アドレス信号と制御信号を分配し、ロングワードのデータとその対応するECC信号をメモリ・ボード信号とバンク・アドレス信号によって選択されたメモリ・バンクに対して転送すると共にこれらがそこから転送されることを可能にする。

いずれのタイプのNVRAM(非揮発性RAM)であってもよいEEPROM640によって、オフ・ライン修理用のメモリ・エラー・データとモジュールのサイズのような構成データが記憶される。故障の発生後メモリ・モジュールが取り外さ

装置612のトランシーバ/レジスタによって、これらの信号をバス80と82との間で授受するバッファとインターフェースが設けられる。ステータス・レジスタ618に記憶されているプライマリ/ミラー・マルチプレクサのビットによって、メモリ制御装置70と75のいずれがプライマリ・メモリ制御装置として指定され、いずれがミラー・メモリ制御装置として指定されかが指示され、プライマリ/ミラー・マルチプレクサ信号がステータス・レジスタ618から指示装置612に加えられる。

プライマリ/ミラー指示装置612によって、制御ロジック610に分配する2組の信号が与えられる。1組の信号は指定されたプライマリ・メモリ・ボード・アドレス、バンク・アドレス、行および列アドレス、サイクル・タイプ、サイクル・タイミングおよびECC信号を含む。他方の組の信号は、指定されたミラー・メモリ・ボード・アドレス信号、バンク・アドレス信号、列および行アドレス信号、サイクル・タイプ信号、サイク

ル・タイミング信号、およびECC信号を含む。プライマリ／ミラー・マルチプレクサ信号は、バス80と82の信号がそれぞれ指定されたプライマリ信号を搬送する線および指定されたミラー信号を搬送する線に向けられるか、またはその逆であるかを選択するために指示装置612によって使用される。

バス80と82には多数の時間分周多重化双方向線が含まれている。メモリ転送サイクル、ステータス・レジスタ転送サイクル、およびEEPROM転送サイクルの開始後一定の時間に、データ・バス75のデータに対応するECC信号がこれらの時間分周多重化双方向線に載置される。もしこの転送サイクルが書き込みサイクルであれば、メモリモジュール60はメモリ制御装置からデータとECC信号を受け取る。もしこの転送サイクルが読み出しサイクルであれば、メモリ・モジュール60はデータとECC信号をメモリ・モジュールに転送する。転送サイクルの他の時間に、アドレス信号、制御信号およびタイミング信号は時間分

周多重化双方向線でメモリ・モジュール60によって受け取られる。メモリ転送サイクル、ステータス・レジスタ転送サイクル、およびEEPROM転送サイクルの始めに、メモリ制御装置70と75がメモリ・ボード・アドレス、バンク・アドレス、およびサイクル・タイプ信号をこれらの時間共有線で各メモリ・モジュール60に転送することが望ましい。

行アドレス信号と列アドレス信号は同じ転送サイクル中に同じ行および列アドレス線で多重化されることが望ましい。先ず、行アドレスがメモリ制御装置によってメモリ・モジュール60に加えられ、約60ナノ秒後に列アドレスが加えられる。シーケンサ616は、システム・クロック信号とリセット信号をCPUモジュール30から入力として受け取り、指定されたプライマリ・サイクル・タイミング信号、指定されたプライマリ・サイクル・タイプ信号、指定されたミラー・サイクル・タイミング信号、および指定されたミラー・サイクル・タイプ信号を指定装置612の

トランシーバ／レジスタから受け取る。

シーケンサ616は、種々のタイプのサイクルを実行するために必要な多数の制御およびシーケンス・タイミング信号を発生し、これらをメモリ・モジュールに対して分配する関連したステアリング(steering)論理を有するリング・カウンタである。制御およびシーケンス・タイミング信号は、システム・クロック信号、指定されたプライマリ・サイクル・タイミング信号、および指定されたプライマリ・サイクル・タイプ信号から発生される。

シーケンサ616は、またシステム・クロック信号、指定されたミラー・サイクル・タイミング信号、および指定されたミラー・サイクル・タイプ信号から重複した組のシーケンス・タイミング信号を発生する。これらの重複したシーケンス・タイミング信号は、エラーのテックのために使用される。高速ページモードで多重のロングワードのデータをメモリ・モジュール60との間で授受するためには、各組の列アドレスは第1の組でス

タートし、127秒遅れて次の列アドレス120がこれに続き、各々のロングワードのデータは前のロングワードのデータの後に120ナノ秒遅れてバス85を横切って移動される。

シーケンサ616は、またtx/rxレジスタ制御信号を発生する。tx/rxレジスタ制御信号は、トランシーバ／レジスタ620の動作と指定装置612のトランシーバ／レジスタを制御するために発生される。データの流れの方向は、シーケンサ616のステアリングロジックによって決定され、このシーケンサ616はtx/rx制御信号とシーケンス・タイミング信号を発生することによって、指定されたプライマリ・サイクル・タイプ信号にตอบสนองし、データとECC信号がメモリ・モジュール60のトランシーバ／レジスタに対して書き込まれるべきであるかまたはここから読み出されるべきであるか、およびそれらが何時行われるべきであることを示す。メモリ書き込みサイクル中、ステータス・レジスタ書き込みサイクル中、および書き込みサイクル中、データお

よびECC信号はバス80、82、および85からトランシーバ/レジスタにラッチされ、一方メモリ読み出しサイクル中、ステータス・レジスタ読み出しサイクル中、およびEEPROM読み出しサイクル中、データおよびECC信号は、メモリ・アレイ600、ステータス・レジスタ618、またはEEPROM640からトランシーバ/レジスタにラッチされて、CPUモジュール30に出力される。

シーケンサ616は、またEEPROM制御信号を発生して、EEPROM640の動作を制御する。

メモリ・モジュール60に存在するタイミング関係はシステム・クロック信号の立ち上がり時間を参考にして決められるが、このシステム・クロック信号は30ナノ秒の間隔を有している。全てのステータス・レジスタ読み出しおよび書き込みサイクルと1つのロングワードの全てのメモリ読み出しおよび書き込みサイクルは、10システム・クロックの時間内、すなわち300ナノ秒内に

る。

CPUモジュール30によって要求されたサイクルが実行されるに従って、シーケンサ616によって発生されたシーケンス・タイミング信号はシーケンサの入力した異なった状態と関連する。これらの異なった状態の間のタイミング関係（およびこれらの状態の各々に対応するシーケンス・タイミング信号の間のタイミング関係）を決めるため、シーケンサ616によって入力することのできるディスクリートな状態がSEQ IDLEおよびSEQ 1ないしSEQ 19と識別される。各状態は、1システム・クロックの間隔（30ナノ秒）の間持続する。シーケンサ616の行う各々の異なった状態に対する入力は、システム・クロック信号の立ち上がり区間によってトリガされる。シーケンサ616に状態SEQ IDLEおよびSEQ 1ないしSEQ 19を入力させるシステム・クロック信号の立ち上がり区間は、これらをシーケンサ616の状態と関連させるために遷移T IDLEおよびT 1ないしT 19として表さ

実行される。メモリ読み出しおよび書き込み転送サイクルは、多重化されたロングワードの転送によって構成されることができる。別のロングワードが転送される毎に、メモリ転送サイクルは4システム・クロックの期間だけさらに延長される。メモリ・リフレッシュ・サイクルとEEPROM書き込みサイクルを実行するには少なくとも12システム・クロックの間隔が必要であり、EEPROM読み出しサイクルは、少なくとも20システム・クロックの間隔を必要とする。

指定されたプライマリ・サイクル・タイミング信号によって、シーケンサ616はシーケンス・タイミング信号と制御信号との発生を開始し、これらの信号によって、メモリ・ポート・アドレス信号によって選択されたメモリ・モジュールが要求されたサイクルを実行することが可能になる。指定されたプライマリ・サイクル・タイミング信号が活性状態に遷移すると、サイクルが開始される。指定されたプライマリ・サイクル・タイミング信号が不活性状態に戻ると、サイクルは終了す

れる。すなわち、TNはシーケンサ616に状態SEQ Nを入力させるシステム・クロック信号の立ち上がり区間である。

CPUモジュール30がメモリ・モジュール60に1つのサイクルを実行させていない場合、指定されたプライマリ・サイクル・タイミング信号は表明されず(not asserted)、シーケンサはSEQ IDLEの状態のままである。もし制御ロジック610とシーケンサ616がこれもまたバス80でメモリ制御装置70から転送されたメモリ・ボード・アドレスによって選択されたメモリ・モジュールに位置しているならば、シーケンサはメモリ制御装置70によるバス80のサイクル・タイミング信号の表明にตอบสนองしてスタートされる（状態SEQ 1を入力する）。指定されたプライマリ・サイクルの活性信号の表明に続く第1システム・クロック信号の立ち上がり区間は、遷移T 1に対応する。

前に述べたように、メモリ・アレイ600に対して1つのロングワードを授受する場合、そのサ

イクルは10システム・クロックの間隔で実行される。シーケンサはSEQ IDLEから状態SEQ 1ないしSEQ 9に進み、SEQ IDLEに戻る。

しかし、別のロングワードを転送するためにメモリ読み出しおよび書き込みサイクルを延長することができる。メモリ・アレイ600は、「高速ページ・モード」DRAMを使用することが望ましい。多重化されたロングワードの読み出しおよび書き込みを行う期間中、最初のロングワードの転送の後に行われるメモリ・アレイとのデータの授受は、列アドレスを繰り返して更新し、CAS（列アドレス・ストロブ）信号を再び発生することによって行われる。

多重化されたロングワードの転送サイクルの期間中、これらの列アドレスのこれらの更新を実行することが可能であるが、その理由は、全てのロングワードが転送されるまでシステム616は、SEQ 4からSEQ 7を繰り返して循環するからである。例えば、もし3つのロングワードがメモ

リ・アレイ600から読み出され、またはこれに書き込まれているならば、シーケンサは状態SEQ IDLE、SEQ 1、SEQ 2、SEQ 3、SEQ 4、SEQ 5、SEQ 6、SEQ 7、SEQ 4、SEQ 5、SEQ 6、SEQ 7、SEQ 4、SEQ 5、SEQ 6、SEQ 7、SEQ 8、SEQ 9、およびSEQ IDLEを入力する。

メモリ転送サイクルの期間中、指定されたプライマリ・サイクル・タイミング信号は、遷移T6の間のシーケンサ616によって監視され、少なくとも1つの別のロングワードを転送するため、メモリ読み出しまたは書き込みサイクルを延長すべきかどうかを決定する。指定されたプライマリ・サイクル・タイミング信号が遷移T6中に表明された場合、状態SEQ 7にあるシーケンサは状態SEQ 8を入力する代わりに状態SEQ 4を入力することによって、次のシステム・クロック信号に応答する。

多重ロングワードの転送の場合、指定されたプライマリ・サイクル・タイミング信号は、少な

くとも第1T6の遷移の15ナノ秒前に表明され、最後のロングワードが転送される迄、表明されたままである。最後のロングワードが転送されてしまった後でメモリ転送サイクルを終了するため、指定されたプライマリ・サイクル・タイミング信号が最後のT6の遷移の少なくとも15ナノ秒前に表明を解かれ、最後のT6の伝送の後、少なくとも10ナノ秒間表明を解かれたままになる。

メモリ転送サイクルの期間中、指定されたプライマリ行アドレス信号と指定されたプライマリ列アドレス信号は、制御ロジック610内の指定装置612によって別の時間に1組の時間分周多重化線上でメモリ・ドライバ630に与えられる。ドライバ630の出力はメモリ・アレイ600のDRAMのアドレス入力に加えられ、また指定されたミラー行および列アドレス信号と比較するため制御ロジック610に戻されてエラーをテックする。状態レジスタ転送サイクルとEEPROM転送サイクルの期間中、列アドレス信号は指定の記憶場所を選択するために必要ではない。

メモリ転送サイクルの期間中、行アドレス信号はバス80と82の時間を共有する行および列アドレスに与えられた最初の信号である。状態SEQ IDLEの期間中、行アドレス信号は、メモリ制御装置によって行および列アドレス線で転送され、列アドレスはT1の遷移の少なくとも15ナノ秒前からT1の遷移後の10ナノ秒まで安定した状態にある。次に、列アドレス信号はメモリ制御装置によって行およびコラムアドレス線で転送され、列アドレスは、T3の遷移の10ナノ秒前からT4の遷移の後15ナノ秒まで安定した状態にある。メモリ転送サイクルの期間中に多重ロングワードの転送を行う場合、続いて発生する列アドレス信号は、次に行およびコラムアドレス線で転送され、これらの続いて発生する列アドレスはT6の遷移の10ナノ秒前からT7の遷移の後15ナノ秒まで安定した状態にある。

ジェネレータ/チエッカ617はシーケンサ616によって発生された2組のシーケンス・タイミング信号を受け取る。更に、指定されたブラ

イマリ・サイクル・タイプ信号とバンク・アドレス信号および指定されたミラー・サイクル・タイプ信号とバンク・アドレス信号が指定装置612によってジェネレータ/チェッカ617に転送される。ジェネレータ/チェッカでは、多数のプライマリ制御信号、すなわちRAS（行アドレス信号）、CAS（列アドレス・ストロブ）およびWE（書き込みイネーブル）が発生され、プライマリ・シーケンス・タイミング信号と指定されたプライマリ・サイクル・タイプ信号およびバンク・アドレス信号を使用してドライバ630に分配される。これらの制御信号の重複した組がジェネレータ/チェッカ617によって重複（ミラー）シーケンス・タイミング信号と指定されたミラー・サイクル・タイプ信号およびバンク・アドレス信号から発生される。これらのミラーRAS、CAS、および書き込みイネーブル信号はエラーのチェックのために使用される。

プライマリ・サイクル・タンブ信号がメモリ転送サイクルが実行中であることを示す場合、プラ

イマリ・バンク・アドレス信号はメモリ・アレイ600内のDRAMの1つの選択されたバンクを識別する。メモリ・ドライバ630はメモリ・アレイ600内のDRAMの各バンクに対して別々のRASドライバを有している。ジェネレータ/チェッカ617においてプライマリRAS信号は、メモリ転送サイクル中に発生され、ジェネレータ/チェッカをRASドライバに接続する線の1つに非多重化される。その結果、選択されたDRAMバンクに対応するRASドライバのみがメモリ転送サイクル中に表明されたRAS信号を受け取る。リフレッシュ・サイクルの期間中、プライマリRAS信号は非多重化されず、表明されたRAS信号が各RASドライバによって受け取られる。ステータス・レジスタ転送サイクルとEEPROM転送サイクルの期間中、バンク・アドレス信号は不必要である。

メモリ・ドライバ630はまたCASドライバを有する。ジェネレータ/チェッカ617において、プライマリCAS信号はメモリ転送サイクル

とリフレッシュ・サイクルの期間中に発生される。プライマリCAS信号は、非多重化されず、表明されたCAS信号は各CASドライバによって受け取られる。

メモリ書き込みサイクルの期間中、プライマリWE信号はジェネレータ/チェッカ617によって発生される。表明されたWE信号は、ドライバ630によってメモリ・アレイ600内の各DRAMバンクに加えられる。しかし、書き込みは選択されたDRAMバンクによってのみ実行することが可能であり、このDRAMバンクはまた表面されたRASおよびCAS信号を受け取る。

本発明の好適な実施例の場合、メモリ転送サイクルの期間中、プライマリRAS信号はT2の遷移の期間中表明され、T3の遷移の少なくとも10ナノ秒前から安定し、最後のT7の遷移の期間中表面を解かれる。プライマリCAS信号は、各T4の遷移の前15ナノ秒間表明され、各T7の遷移の期間中表明を解かれる。メモリ書き込みサイクルの期間中、プライマリWE信号は、T3

の遷移の期間中表明され、最初のT4の遷移の少なくとも10ナノ秒前から安定し、最後のT7の遷移に期間中表面を解かれる。

プライマリ・サイクル・タイプ信号がメモリ・リフレッシュ・サイクルが実行中であることを示す場合、メモリ・アレイ600はシーケンス616によって与えられるプライマリ・シーケンス・タイミング信号に回答して、ジェネレータ/チェッカ617によってメモリ・リフレッシュ動作を実行される。これらのリフレッシュ動作の期間中、RAS信号およびCAS信号が逆の順序でジェネレータ/チェッカによって発生されて分配される。このリフレッシュ・モードはバンク、行または列に対する外部アドレスを必要としない。

転送サイクルの期間中、データがバス85に転送されている時間に、ECC信号はバス80と82の時間分周多重化双方向線で転送される。しかし、これらの同じ線は転送サイクル中の他の時間に制御信号（例えば、サイクル・タイプ）およびアドレス信号（例えば、メモリ・ボード・アド

レスおよびバンク・アドレス) 信号を転送するために使用される。

プライマリ/ミラー指定装置612内のトランシーバ/レジスタはシーケンサ616によって加えられるシーケンス・タイミング信号ともtx/rxレジスタ制御信号にตอบสนองする受信機と発信機を有する。シーケンス・タイミング信号とtx/rxレジスタ制御信号は、バス80と82の時間分周多重化双方向線でECC信号とアドレス信号および制御信号を多重化することを可能にする。

サイクル・タイプ信号、メモリ・ボード・アドレス信号およびバンク・アドレス信号のような制御信号とアドレス信号はメモリ制御装置70と75によって転送され、単一の転送サイクルまたは多重ロングワード転送サイクルのいずれかの開始時にバス80と82の時間を共有した線に与えられることが望ましい。これらの信号はサイクル・タイミング信号の活性化と同時に遷移を開始し(シーケンサはSEQ IDLE状態にあるが)、T2の間安定状態にある。従って、指定装置612

のトランシーバ/レジスタにおいて、受信機は起動され、送信機は少なくとも状態SEQ2の終わりと迄そのトリステート・モードにセットされる。

サイクル・タイプ信号は、下記にリストアップした機能、すなわちメモリの読み出し、メモリの書き込み、ステータス・レジスタの読み出し、ステータス・レジスタの書き込み、EEPROMの読み出し、EEPROMの書き込み、およびリフレッシュのいずれがサイクル期間中にメモリ・アサイ60によって実行されるかを識別する。指定装置612によって受け取られた指定されたプライマリ・サイクル・タイプ信号は、シーケンサ616に加えられ、tx/rx制御信号とシーケンス・タイミング信号を発生する場合に使用される。例えば、データ・トランシーバ/レジスタ620および指定装置612のトランシーバ/レジスタにおいて、受信機は起動され、送信機は書き込みサイクル全体を通してシーケンサ616によってトリステート・モードにセットされる。しかし、読み出し期間中のデータ・トランシーバ/

レジスタ620および指定装置612のトランシーバ/レジスタの場合、受信機はトリステート・モードにセットされ、送信機はサイクル・タイプ信号、メモリ・ボード・アドレス信号およびバンク・アドレス信号がこのサイクルの開始時点で受け取られた後シーケンサ616によって起動される。

適切な実施例の場合、メモリ・アレイ600に対して授受されたデータはエラー検出コード(EDC)を使用して各メモリ・モジュール60内でチェックされることが望ましく、このエラー検出コードはメモリ制御装置70と75によって必要とされるコードと同じであることが望ましい。好適なコードは1ビット修正、2ビット検出のエラー修正コード(ECC)であることが望ましい。

メモリ書き込みサイクルの期間中、メモリ制御装置70は少なくとも1つのロングワードのデータをデータ・バス85で転送し、同時に対応する組のECC信号をバス80で転送する。一方、メモリ制御装置75は第2組のECC信号を転送し、

これらの信号はバス82のデータ・バス85のロングワードとまた対応する。

ここで実施されているように、メモリ書き込みサイクルの期間中、各ロングワードに対するデータとECC信号がデータ・トランシーバ/レジスタ620の受信機に与えられると共に指定装置612のトランシーバ/レジスタの受信機に与えられる。データおよびECC信号はT4の遷移の少なくとも10ナノ秒前に安定しており、T6の遷移後15ナノ秒後迄安定した状態にあり、これらのトランシーバ/レジスタにラッチされる。この時間の間、メモリ制御装置70と75はバス80と82の時間を共有した線にアドレス信号と制御信号を加えない。

メモリ書き込みサイクルの期間中に指定装置612によって受け取られた指定されたプライマリECC信号とトランシーバ/レジスタ620によって受け取られたロングワードのデータは、メモリ・アレイ600の8つのバンクの各々に於けるDRAMのデータ入力とECC発生装置623

に加えられる。発生されたECCは比較器625によって指定されたプライマリ・ECCと比較される。指定されたプライマリ・ECC信号は、また指定されたミラー・ECC信号と共にECC比較器625に加えられる。

ここで実施例されているように、メモリ読み出しサイクルの期間中、少なくとも1つのロングワードのデータと対応する組のECC信号がメモリ・アレイ600から読み出され、データ・トランシーバ/レジスタ620と指定装置612のトランシーバ/レジスタにそれぞれ向けられる。メモリ読み出しサイクルの遷移T7の期間中、各ロングワードに対するデータとECC信号はメモリ・アレイ600から入手可能であり、これらのトランシーバ/レジスタにラッチされる。このデータはECC発生装置623に与えられ、その出力はメモリから読み出されたECCと比較される。

ラッチの後、データおよびECC信号は、データ・トランシーバ/レジスタ620の送信機と指定装置612のトランシーバ/レジスタの送信機

によってデータ・バス85とバス80および82に与えられる。同じECC信号は、指定装置612のトランシーバ/レジスタからメモリ制御装置70とメモリ制御装置75に転送される。データバス85とバス80および82で転送されたデータとECC信号は、T7の遷移の15秒後からこれに続くT6の遷移の5ナノ秒前迄(多重ロングワード転送の場合)またはこれに続くT IDLE遷移の5ナノ秒前迄(単一のロングワードの転送または多重ロングワード転送の最後のロングワードの場合)安定した状態にある。この時間間隔の間、メモリ制御装置70と75は、バス80と82の時間を共有したアドレス信号と制御信号を加えない。データ・トランシーバ/レジスタ620の送信機と指定装置612のトランシーバ/レジスタの送信機は、これに続くT IDLE遷移の期間中、トリステート・モードにセットされる。

比較器614は、制御装置70から発生するアドレス信号制御装置およびタイミング信号を制御装置75から発生するこれらに対応するアドレス

信号、制御信号およびタイミング信号と比較するために設けられる。指定されたプライマリ・サイクル・タイミング信号、サイクル・タイプ信号、メモリ・ボード・アドレス信号、およびバンク・アドレス信号は、指定されたミラー・サイクル・タイミング信号、サイクル・タイプ信号、メモリ・ボードアドレス信号、バンク・アドレス信号、行アドレス信号、および列アドレス信号と共に指定装置612から比較器614に加えられる。指定されたプライマリ行アドレス信号および列アドレス信号はドライバ630の出力から比較器614に加えられる。そこで両方の組の信号が比較される。

もし、メモリ制御装置から発生するアドレス信号、制御信号、およびタイミング信号の間で比較のミスがあれば、比較器614は適当なエラー信号を発生する。第6図に示すように、ボード・アドレス・エラー信号、バンク・アドレス・エラー信号、行アドレス・エラー信号、列アドレス・エラー信号、サイクル・タイプ・アドレス・エラー

信号、およびサイクル・タイミング・エラー信号は比較器から発生することができる。

ジェネレータ/チェッカ617は、指定されたプライマリ・バンク・アドレス信号、サイクル・タイプ信号およびサイクル・タイミング信号を使用してシーケンサ616およびジェネレータ/チェッカ617によって発生されたプライマリ制御信号およびタイミング信号を、指定されたミラー・バンク・アドレス信号、サイクル・タイプ信号およびサイクル・タイミング信号を使用して、発生されたミラー制御信号およびタイミング信号と比較する。2組のシーケンス・タイミング信号は、シーケンサ616によってジェネレータ/チェッカ617に加えられる。プライマリRAS信号、CAS信号、およびWE信号は、ドライバ630の出力からジェネレータ/チェッカ617に加えられる。前に説明したように、ミラーRAS信号、CAS信号およびWE信号はジェネレータ/チェッカによって内部的に発生される。ジェネレータ/チェッカ617は、プライマリRAS信号、

CAS信号、WE信号およびシーケンス・タイミング信号をミラーRAS信号、CAS信号、WE信号、およびシーケンス・タイミング信号と比較する。

もし、シーケンサ616またはジェネレータ/チェッカ617から発生する制御信号およびタイミング信号のいずれかの間に比較のミあれば、ジェネレータ/チェッカは適当なエラー信号を発生する。第6図に示すように、シーケンサ・エラー信号、RASエラー信号、CASエラー信号、およびWEエラー信号はジェネレータ/チェッカ617によって発生することかできる。

エラー信号は、比較器614およびジェネレータ/チェッカ617からアドレス/制御エラー・ロジック621に加えられる。比較器614またはジェネレータ/チェッカ617から受け取ったエラー信号にตอบสนองして、アドレス/制御エラー・ロジック621はアドレス/制御エラー信号をCPUモジュール30に転送し、アドレス信号・制御信号、またはタイミング信号のいずれかの間

で比較のミスが発生したことによる故障を検出したことを示す。アドレス/制御エラー信号は、エラーを処理するためにメモリ制御装置70と75のエラー・ロジックに送られる。アドレス/制御エラー信号をCPUモジュール30にすることによって、CPU/MEM故障が発生するが、これは他のセクションで詳細に論じる。

比較器614およびジェネレータ/チェッカ617からのエラー信号は、またステータス・レジスタ618に加えられる。エラー信号および故障に関連するアドレス信号、制御信号、タイミング信号、データ信号およびECC信号の全ては、一時的にステータス・レジスタに記憶され、エラーの診断と修復を可能にする。

本発明の1つの特徴によれば、32ビットのデータ・バス85が1本だけCPUモジュール30とメモリ・モジュール60との間に設けられる。従って、メモリ・モジュール60はメモリ制御装置70と75からの2組のデータを比較することができない。しかし、メモリ制御装置70と75

によってメモリ・モジュール60に転送された2つの独立した組のECC信号をチェックすることによって、ビットのデータ線の重複した組を使用することなく、データの健全性がメモリ・モジュール60によって検証される。

第6図に示すように、制御ロジック610はECC発生装置623とECC比較器625を有する。指定されたプライマリおよびミラーECC信号は、指定装置712によってECC比較器に加えられる。メモリー書き込みサイクルの期間中、指定されたプライマリECC信号は、指定されたミラーECC信号と比較される。その結果、メモリ・モジュール60は、メモリ制御装置70と75が一致しているかどうかを検証すると共にメモリー書き込みサイクルの期間中にメモリ・アレイ600のDRAMに記憶されている指定されたプライマリECC信号が正しいかどうかを検証する。更に、メモリー書き込みサイクルの期間中にDRAMのデータ入力に与えられたデータは、ECC発生装置623に加えられる。ECC発生

装置623は、このデータ応する1組の発生されたECC信号を発生し、この発生されたECC信号をECC比較器625に加える。指定されたプライマリECC信号は発生されたECC信号と比較され、メモリ制御装置70によってデータ・バス85に転送されたデータがメモリ・アレイ600のDRAMに記憶されているデータと同じであるかどうかを検証する。

メモリー読み出しサイクルの期間中、DRAMの選択されたバンクから読み出されたデータはECC発生器に与えられる。発生されたECC信号はそこでECC比較器に加えられ、このECC比較器は、またDRAMの選択されたバンクから読み出されて記憶されているECC信号を受け取る。発生され記憶されているECC信号は、ECC比較器625によって比較される。

もしECC比較器625によって監視されているECC信号のいずれかの対の間に比較のミスがあれば、ECC比較器は適当なエラー信号を発生する。第6図に示すように、プライマリ/ミラー

ECCエラー信号、プライマリ／発生されたECC信号エラーおよびメモリ／発生されたECCエラー信号はECC比較器によって発生することができる。

ECC比較器625からのこれらのECCエラー信号はステータス・レジスタ618に加えられる。ECCエラー信号の各々およびECC故障に関連するアドレス信号、制御信号、タイミング信号、データ信号、およびECC信号の全ては一時的にステータス・レジスタに記憶され、エラーの診断と修復を可能にする。

ECCエラー信号はECCエラー線上でECC比較器625によって表明され、CPUモジュール30に転送され、比較のミスによって発生したECCの故障を検出したことを示す。この比較のミスはメモリ書き込みサイクルの期間中に行われる2つのECCのチェックの期間中またはメモリ読み出しサイクル行われる1つのECCのチェックの期間中のいずれかで発生する可能性がある。

第6図に示すように、ボード選択ロジック627

はメモリの背面板からスロット信号を受け取る。これらのスロット信号によって、各メモリモジュール60に対してユニークなスロット・ロケーションが指定される。ボード選択ロジック627は、そこでこれらのスロット信号を指定回路612を介してメモリ制御装置の1から転送された指定されたプライマリ・ボード・アドレス信号と比較する。もしこのスロット信号が指定されたプライマリ・ボード・アドレス信号と同じであれば、ボード選択信号がボード選択ロジック627によって発生され、これによって制御ロジック610内の他の回路を動作させる。

3. メモリ制御装置

メモリ制御装置70と75は、CPU40と50のメモリ・モジュール60および補助メモリ素子に対するアクセスをそれぞれ制御好適な実施例の場合、ある種のエラー処理動作を実行する。メモリ制御装置72に接続された補助メモリ素子はシステムROM43、EEPROM44、およびスクラッチ・パッドRAM45を有する。ROM

43は、診断コード、コンソール・ドライバ・コード、およびブートストラップ・コードの一部のようなある種の標準コードを保持している。

EEPROM44は、CPU40の動作中に検出されたエラー情報のような情報を保持するのに使用されるが、この情報は変更を行う必要があるが、電源を切った場合に失われるべきではない。スクラッチ・パッドRAM45は、CPU40によって実行されるある種の動作のために使用されると共に、ルール・ユニーク情報（例えば、ただ1つのCPU40または50に使用することのできる1つのルールの条件に特有の情報）をゾーン情報（CPU40と50の両方がアクセスすることのできる情報）に変換するために使用される。

等価な構成要素53、54および55がメモリ制御装置75に接続される。システムROM53、EEPROM54およびスクラッチ・パッドRAM55は、システムROM43、EEPROM44、およびスクラッチ・パッドRAM45とそれぞれ同じであり、同じ機能を実行する。

第7図ないし第9図は、プライマリ・メモリ制御装置70の好適な実施例の詳細を示す。ミラー・メモリ制御装置75は、第7図ないし第9図に示す構成要素と同じ構成要素を有しているが、動作は若干異なっている。従って、メモリ制御装置75の動作と異なっている部分を除いて、プライマリ・メモリ制御装置70の動作のみを説明する。処理システム20'内のメモリ制御装置70'と75'は同じ構成要素を有し、それぞれメモリ制御装置70と75と同じように動作する。

第7図に示す構成要素は、プライマリ・メモリ制御装置70を介してデータの流れ、アスおよび信号を制御する。制御ロジック700は、メモリ制御装置70の受け取った信号および制御ロジック700に記憶されているそのメモリ制御装置のステート・エンジンに従って第7図の種々の構成要素の状態を制御する。マルチプレクサ702は、これらのソースの1つからアドレスを選択する。これらのアドレスは、受信機705を介してCPU30から得ることもできるし、第8図を参照して

以下で説明するDMAエンジン800から得ることもできるし、また再同期化動作の期間中に1つのゾーンから他のゾーンにある種のバンク・メモリを転送する間に人工的リフレッシュを発生するのに使用されるリフレッシュ再同期化アドレスから得ることもできる。

CPU30からのデータは、受信機705を介して受け取られDMAからのデータはエンジン800を介して受け取られるので、マルチプレクサ702の出力はマルチプレクサ0の入力である。マルチプレクサ710の出力は、メモリ相互接続部85とドライバ715を介してメモリ・モジュール60にデータを与える。ドライバ715はミラー・メモリ制御モジュール75と75'に対して不能にされるが、その理由は、メモリ・データの1つの組のみが、それぞれメモリ・モジュール60と60'に送られるからである。

メモリ相互接続部85に送られるデータは、CPU30からメモリ・モジュール60に記憶されるべきデータまたはDMAエンジン800から

メモリ・モジュール60に記憶されるべきデータのいずれかを含んでいる。CPU30からのデータとマルチプレクサ702からのデータはまたこの経路また受信機745とECC修正装置750を介してDMAエンジン800に送られる。

マルチプレクサ702からのアドレスは、デマルチプレクサ720の入力にまた加えられ、このデマルチプレクサ720はこれらのアドレスを行／列アドレス部、ボード／バンク・アドレス部分およびシングル・ボード・ビットに分割する。行／列アドレスの22ビットが11本の線に多重化される。好適な実施例の場合、22ビットの行／列アドレスがドライバ21を介してメモリ・モジュール60に送られる。シングル・ボード・ビットはドライバ722を介してメモリ・モジュール60に送られることが望ましく、他のボード／バンク・アドレスビットはECC信号と多重化される。

マルチプレクサ725は、メモリ制御装置70に対する通常のリフレッシュ命令とCPU30か

らのサイクル・タイプ情報(すなわち読み出し、書き込み等)およびDMAサイクル・タイプ情報とを結合する。通常のリフレッシュ命令とリフレッシュ再同期アドレスの両方によって、メモリ・モジュール60がメモリ・リフレッシュ動作を開始する。

マルチプレクサ725の出力は、デマルチプレクサ720からのボード／バンク・アドレスと共にマルチプレクサ730に対する入力である。マルチプレクサ730に対する他の入力は、ECCジェネレータ／チェッカ735の出力である。マルチプレクサ730は、入力の1つを選択し、これをメモリ・モジュール60に対する時間分割多重化ECC／アドレス線に載置する。マルチプレクサ730は、これらの時間分割多重化線がボード／バンク・アドレスと別の制御情報ならびにECC情報を、異なった時間に、搬送することを可能にする。

ECC情報は、受信機734を介してメモリ・モジュール60から受け取られ、入力としてECC

ジェネレータ／チェッカ735に加えられ、メモリ・モジュール60によって発生されたECCをメモリ制御装置70によって発生されたECCと比較する。

ECCジェネレータ／チェッカ735に対する他の入力は、マルチプレクサ740からの出力である。メモリ・トランザクションが書き込みトランザクションであるか読み出しトランザクションであるかによって、マルチプレクサ740はマルチプレクサ710からメモリ・モジュール60に送られたメモリ・データを入力として受け取るか、または受信機745を介してメモリ・モジュール60から受け取られたメモリ・データを入力として受け取る。マルチプレクサ740は、ECCジェネレータ／チェッカ735に対する入力であるこれらのメモリ・データの組の1つを選択する。ジェネレータ／チェッカ735は、次に適当なECCコードを発生し、このコードは、マルチプレクサ730に送られる以外に、またECC修正装置750にも送られる。好適な実施例の場合、

ECC修正装置750はメモリ・モジュール60から受け取られたメモリ・データ内の全てのシングル・ビットエラーを修正する。

ECCチェック750からの修正されたメモリ・データは、次に第8図に示すDMAエンジンに送られると共にマルチプレクサ752に送られる。マルチプレクサ752に対する他の入力、第9図と関連して以下で説明するエラー処理ロジックからのエラー情報である。マルチプレクサ752の出力は、ドライバ753を介してCPU30に送られる。

比較器755は、マルチプレクサ710からメモリ・モジュール60に送られたデータをこのデータがドライバ715と受信機745を通過した後、このデータのコピーと比較する。チェックによって、ドライバ715と受信機745が正しく動作しているかどうかを判定する。比較器755からの出力はCMPエラー信号であり、この信号はこのような比較エラーがあるか無いかを示す。第9図においてCMPエラー・ロジックに供給さ

れる。

第7図の他の2つの構成要素によって、異なる種類のエラー検出が行われる。構成要素760はパリティ発生装置である。メモリ制御装置70によってメモリ・モジュール60に記憶されるべきデータに発生されたかまたはメモリ・モジュール60によってメモリ・モジュール60から読み出されたデータに発生されたECCデータは、パリティ発生装置70に送られる。発生装置760からのパリティ信号は、ドライバ762を介して、比較器765に送られる。比較器765は、発生装置760からECCパリティ信号を制御装置75'によって発生された等価のECCパリティ信号と比較する。

パリティ発生装置770は、デマルチプレクサ720から受け取られた行/列アドレス信号とシングル・ビット・ボード・アドレス信号とについて同じ種類のチェックを実行する。パリティ発生装置770からのアドレス・パリティ信号はドライバ772によって比較器775に送られ、

この比較器775は制御装置75からまたアドレス・パリティ信号を受け取る。比較器765と775の出力はパリティ・エラー信号であり、これらの信号は第9図のエラー・ロジックに供給される。

第8図はDMAエンジン800の基礎を示す。好適な実施例の場合、DMAエンジン800はメモリ制御装置70内に位置するが、この場所にある必要はない。第8図に示すように、DMAエンジン800はデータ・ルータ(router)810、DMA制御装置820、およびDMAレジスタ830を有する。ドライバ815と受信機816によって、メモリ制御装置70とクロスリンク90との間にインターフェースが設けられる。

DMA制御装置820は、制御ロジック700から内部制御信号を受け取り、これにตอบสนองして、制御信号を送ってデータ・ルータ810を適当に構成する。制御装置820によって、データ・ルータ810が第7図に示すクロスリンク90からのデータと制御信号をメモリ制御70回路に送る

ように、その構成がまた設定される。データ・ルータは、その状態信号をDMA制御装置820に送り、このDMA制御装置はこの信号を他のDMA情報と共に第9図のエラー・ロジックに伝える。

レジスタ830はDMAバイト・カウンタ・レジスタ832とDMAアドレス・レジスタ836を有する。これらのレジスタは、ルータ810を介してCPU40によって初期値にセットされる。次に、DMAサイクルの期間中、制御装置820はルータ810を介してカウンタ・レジスタ832をインクリメントさせアドレス・レジスタ836をデクリメントさせる。制御装置820によって、アドレス・サイクル836の内容がDMA動作の期間中ルータ810と第7図の回路を介してまたメモリ・モジュール60に送られる。

上に説明したように、本発明の好適な実施例の場合、メモリ制御装置70、75、70'、および75'は、またある種の基本的なエラー動作を実行する。第9図は、このようなエラー動作を実行するハードウェアの好適な実施例の1例を示す。

第9図に示すように、タイムアウト信号、ECCエラー信号およびバスのミス比較信号のようなある種のメモリ制御装置内部信号は、レール (rail) ・エラー信号、ファイヤーウォール (firewall) のミス比較信号およびアドレス／制御エラー信号のようなある種の外部信号と同様に、診断エラー・ロジック870に対する入力である。好適な実施例の場合、診断エラー・ロジック870はクロスリンク90と95を介してシステム10の他の構成要素からエラー信号を受け取る。

診断エラーロジック870は、エラー信号とメモリ制御装置70のベシック・タイミングから発生された制御パルス信号からエラー・パルスを形成する。診断エラー・ロジック870によって発生されたエラー・パルスは、ある種のタイミング信号に従って診断エラー・レジスタ880の適当なロケーションに記憶されているある種のエラー情報を含む。システム故障エラー・アドレス・レジスタ65は、エラーが発生した場合、CPU40と50が通信を行っていたメモリ・モジュール

60内にアドレスを記憶する。

診断エラー・ロジック870からのエラー・パルスはまたエラー・カテゴリー化ロジック850に送られ、このエラー・カテゴリー化ロジック850はまたサイクル・タイプ (例えば読み出し、書き込み等) を示す情報をCPU30から受け取る。この情報およびエラー・パルスから、エラー・カテゴリー化ロジック850はCPU/I Oエラー、DMAエラー、またはCPU/MEM故障の存在を判定する。

CPU/I Oエラーは、バス46のCPU/I Oサイクルに直接帰すべき動作上のエラーであり、リセットに関して以下で説明するように、ハードウェアによって修復することが可能である。DMAエラーは、DMAサイクルの期間中に発生するエラーであり、好適な実施例の場合、主としてソフトウェアによって処理される。CPU/MEM故障は、CPUの正しい動作またはメモリの内容を保障することのできないエラーである。

エラー・カテゴリー化ロジック850からの出

力は、エンコーダ855に送られ、このエンコーダ855は特定のエラー・コードを形成する。このエラー・コードは、エラー・ディスエーブル信号が存在する場合、次にANDゲート856を介してクロスリンク90と95に送られる。

エラー・コードを受け取った後、クロスリンク90、95、90'、95'はメモリ制御装置にリトライ要求信号を送る。第9図に示すように、メモリ制御装置70のエンコーダ895はサイクル・タイプ情報とエラー信号 (サイクル・クオリファイヤ (qualifiers) として纏めてに示される) と共にリトライ要求信号を受け取る。エンコーダ895は、次にシステム故障エラー・レジスタ898に記憶するための適当なエラー・コードを発生する。

システム故障エラー・レジスタ898は、診断エラー・レジスタ880と同じ情報を記憶しない。システム故障エラー・レジスタ898とは違って、診断エラー・レジスタ880はクロスリンク・レールからの1つの入力のエラーのようなレール・

ユニーク情報およびメモリ・モジュール60内の修正不可能なECCエラーのようなゾーン・ユニーク・データのみを含んでいる。

診断エラー・レジスタ898は、またエラーの処理に使用される幾つかのビットを含んでいる。これらのビットは、所望のメモリ・ロケーションが見当たらないことを示すNXNビット、所望にI/Oロケーションが見当たらないことを示すNXI Oビット、ソリッド故障ビットおよび過渡的ビットを含んでいる。過渡的ビットソリッド・ビットはいずれも故障のレベルを示す。過渡的ビットによって、またシステム故障エラー・アドレス・レジスタ865が凍結される。

第9図は、メモリ・コントローラ・ステータス・レジスタ875を示すが、これは技術的にはエラー・ロジックの一部ではない。レジスタ875は、DMA比率部877のDMA比率コード・エラー・ディスエーブル部878のエラー・ディスエーブル・コード、およびミラー・バス・ドライバ・イネーブル部876のミラー・バス・ドライバ

・イネーブルコードのようなある種の状態情報を記憶する。DMA比率コードは、DMAに割り当てることができるメモリ帯域幅の部分特定する。エラー・デスエーテル・コードによって、ANDゲート856および従ってエラー・コードを不能にする信号が与えられる。ミラー・バス・ドライバ・イネーブル・コードによって、ある種のトランザクションに対してミラー・バス・ドライバを動作させる信号が与えられる。

4. クロスリンク

メモリ再同期、DMAおよびI/O動作のデータは、クロスリンク90と95を通過する。一般的に、クロスリンク90および95によって、CPUモジュール30、CPUモジュール30'、I/Oモジュール100、110、120、およびI/Oモジュール110'、110'、120'、との間の通信が行われる。(第1図参照)

クロスリンク90と95は、第10図に示すように、並列レジスタ910と直列レジスタ920の両方を含む。両方のタイプのレジスタは、本発

明の好適な実施例でプロセッサ間の通信を行うために使用される。通常の動作の期間中、処理システム20と20'は同期され、データはそれぞれクロスリンク90/95と90'/95'の並列レジスタ910を使用して、処理システム20と20'との間で交換され、処理システム20と20'が同期されていない場合、ブートストラッピングの期間中に最も顕著に現れるように、データは直列レジスタ902によってクロスリンクの間に交換される。

並列レジスタのアドレスは、メモリ・スペースと違ってI/Oスペースである。メモリ・スペースとはメモリモジュール60内のロケーションのことである。I/Oスペースとは、I/Oおよび内部システム・レジスタのようなロケーションのことであり、これはメモリ・モジュール60内には存在しない。

I/Oスペース内では、アドレスはシステム・アドレス・スペース内に存在するか、ゾーン・アドレス・スペース内に存在するかのいずれかであ

る。「システム・アドレス・スペース」という用語は、システム10全体を通してアクセスすることのできるアドレス、すなわち処理システム20と20'の両方によってアクセスすることのできるアドレスのことである。「ゾーン・アドレス・スペース」という用語は、特定のクロスリンクを含むゾーンによってのみアクセス可能であるアドレスのことである。

第10図に示す並列レジスタは、通信レジスタ906とI/Oリセット・レジスタ908を有する。通信レジスタ906は、ゾーン間で交換される独特のデータを含む。このようなデータは、メモリ・ソフト・エラーのような通常ゾーンに特有のデータである(メモリモジュール60と60'が同じエラーを同時に独立して経験するということは確率の領域外の出来事である)

レジスタ906に記憶されるべきデータはユニークなものであるため、書き込みの目的のための通信レジスタ906のアドレスは、ゾーン・アドレス・スペースになければならない。もしそうで

なければ、処理システム20と20'は、ロックステップ同期状態にあり同じ連の命令を同時に実行しているため、ゾーン・ユニーク・データをゾーン11内の通信レジスタ906のみに記憶することはできず、これらはこの同じデータをゾーン11'内の通信レジスタ906'(図示せず)にも記憶しなければならない。

しかし、読み出しのための通信レジスタ906のアドレスは、システム・アドレス・スペース内に存在する。したがって、同期動作の期間中、両方のゾーンは同時に1つのゾーンから通信レジスタを読み出すことができ、次に他のゾーンから通信レジスタを同時に読み出すことができる。

I/Oリセット・レジスタ908は、システム・アドレス・スペース内に存在する。このI/Oリセット・レジスタは、対応するモジュールがリセット状態にあるかどうかを示すため、1つのI/Oモジュールに対して1ビットを有する。I/Oモジュールがリセット状態にある場合、これは効果的にディスエーブルされる。

並列レジスタ91もまた他のレジスタを有するが、これらの他のレジスタの理解は本発明を理解するために必要ではない。

並列クロスリンク・レジスタ920は全てゾーンの固有のスペース内に存在するが、その理由は、これらが非同期通信に使用されるが、ゾーンに固有の情報のみを有しているかのいずれかであるからである。並列クロスリンク・レジスタと並列クロスリンクの目的は、プロセッサ20と20'が例えロック・ステップ同期状態（例えば、位相ロック状態およびこれと同じ状態）で動作していても、これらのプロセッサ20と20'に通信を行なわせることである。好適な実施例の場合、幾つかの並列レジスタがあるが、本発明を理解するためにこれらを説明する必要はない。

制御および状態レジスタ912は、状態および制御フラグを含む直列レジスタである。これらのフラグの1つはOSRビット913であり、これはブートストラピングのために使用され、対応するゾーンの処理システムがブートストラブ・プロ

セスが既に終了しているかまたはこのシステムが再同期を行ったかのいずれかの理由のために、この処理システムが既にそのブートストラブ・プロセスを開始したかどうか、またはそのゾーンに対する動作システムが現在動作中であるかどうかを示す。

制御および状態レジスタ912は、またクロスリンク90の現在のモードおよび従って処理システム20の現在のモードを識別するためのモード・ビット914を有する。モード・ビットは、再同期モード・ビット915とクロスリンク・モード・ビット916を含むことが望ましい。再同期モード・ビット915は、クロスリンク90を再同期スレーブ・モードまたは再同期マスター・モードのいずれかにあるものとして識別する。クロスリンク・モード・ビット916は、クロスリンク90をクロスリンク・オフ・モード、デュプレックス・モード、クロスリンク・マスター・モード、またはクロスリンク・スレーブ・モードのいずれかにあるものとして識別する。

直列レジスタの用途の1つは、状態読み出し動作であり、この動作によって、1つのゾーンのクロスリンクが他のゾーンのクロスリンクの状態を読み出すことができる。状態読み出し要求フラグ918を直列制御状態レジスタ912に立てることによって、状態情報に対する要求がクロスリンク90'に送られる。このメッセージを受け取ると、クロスリンク90'は、その直列制御および状態レジスタ912'の内容をクロスリンク90に送り返す。

第11図は、プライマリ・クロスリンク90およびミラー・クロスリンク95内のルート制御および状態信号（「制御コード」と呼ぶ）用の構成要素の幾つかを示す。対応するクロスリンクの構成要素は、好適な実施例では、クロスリンク90および95'内に存在する。これらのコードは、メモリ制御装置70と75およびモジュール相互接続部130、132、130'および132'との間に送られる。

第12図は、ルート・データおよびアドレス信

号を送るのに使用される好適な実施例のプライマリ・クロスリンク90の構成要素を示す。対応するクロスリンクの構成要素は、クロスリンク95、90'および95'内に存在する。

第11図は、プライマリ・クロスリンク90とミラー・クロスリンク95の両方に対する構成要素を示すが、これらの構成要素の間には重要な相互接続部があるため、ハードウェアは同じである。プライマリ・クロスリンク90の構成要素と同じミラー・クロスリンク95の回路の構成要素は同じ番号で示すが、ミラー制御装置の場合には番号の次に「m」の文字を付ける。

第11図および第12図を参照して、これらの構成要素はラッチ、マルチプレクサ、ドライバおよび受信機を含む。ラッチ933および933mのような一部のラッチは遅延要素として動作し、クロスリンクの正しいタイミングを保証し、これによって同期を維持する。第11図に示すように、メモリ制御装置70からの制御コードは、バス88を介してラッチ931に送られ、次にラッチ

932に送られる。このようなラッチを行う理由は、適当な遅れを与えてメモリ制御装置70からのデータがメモリ制御装置70'からのデータと同時にクロスリンク90を通過することを保証することである。

もしメモリ制御装置70からのコードがクロスリンク90'を介して処理システム20'に送られるべきであれば、ドライバ937が起動される。メモリ制御装置70からの制御コードは、またラッチ933を通過してマルチプレクサCSMUXA935に入る。もし制御コードがクロスリンク90'からプライマリ・クロスリンク90に受け取られれば、これらの経路は受信装置936を通過してラッチ938およびまたマルチプレクサ935に至る。

マルチプレクサ935に対する制御コードによって、データのソースが決定される、すなわちこれがメモリ制御装置70からきたものであるかまたはメモリ制御装置70'からきたものであるかが決定され、これらのコードはマルチプレクサ935の出力に加えられる。この出力は、再び正

しい遅延目的のため、ラッチ939の記憶され、もしこれらのコードがモジュール相互接続部130に送られるべきであれば、ドライバ940が起動される。

データおよびアドレス信号の経路は、第12図に示すように、第11図に示す制御信号の経路と若干類似している。これらの相違点は、いずれの1つのトランザクションの期間中においてもデータおよびアドレスはクロスリンク90と95を介して1つの方向のみに流れるが、制御信号はそのトランザクションの期間中に双方向に流れるという事実を反映している。これと同じ理由のため、バス88と89のデータ線は双方向であるが、制御方向は双方向ではない。

バス88を介してメモリ制御装置70から供給されるデータとアドレスはラッチ961に入り、次いでラッチ962に入り、次いでラッチ964に入る。第11図の場合と同様に、第12図のラッチによって同期を維持するための正しいタイミングが与えられる。メモリ制御装置70'から出

力されるデータは受信装置986によってバッファされ、ラッチ988に記憶され、次にマルチプレクサMUXA966の入力に向かう。マルチプレクサ966の出力は、ラッチ986に記憶され、もしドライバ969が起動されれば、モジュール相互接続部130に送られる。

第11図はメモリ制御装置72送られるべき制御コードの経路を示す。モジュール相互接続部130からのコードは、先ずラッチ941に記憶され、次にマルチプレクサCSMUXC942に与えられる。マルチプレクサ942は、また並列クロスリンク・レジスタ910から制御コードを受け取り、ラッチ943に転送するため並列レジスタ・コードまたはラッチ941からのコードのいずれかを選択する。もしこれらの制御コードがクロスリンク90'に転送されるべきであれば、ドライバ946が起動される。クロスリンク90'からのコード（および従ってメモリ制御装置70'からの制御コード）は受信機947によってバッファされ、ラッチ948に記憶され、入力として

マルチプレクサCSMUXD945に加えられる。マルチプレクサCSMUXD945は、またラッチ943の内容を記憶しているラッチ944の出力を入力として受け取る。

マルチプレクサ945は、モジュール相互接続部130からのコードまたはクロスリンク90'からのコードのいずれかを選択し、これらの信号を入力としてマルチプレクサCSMUXE949に加える。マルチプレクサ949は、またデコード・ロジック970からのコード（再同期の期間中に発生するバルク・メモリの転送のために）、直列クロスリンク・レジスタ920からのコード、または所定のエラーコードERRを入力として受け取る。マルチプレクサ949は、次に適当に制限されてこれらの入力の幾つかを選択してラッチ950に記憶する。もしこれらのコードがメモリ制御装置70に送られるべきであれば、次にドライバ951が起動される。

マルチプレクサ949に対する入力であるエラー・コードERRの目的は、レールの1つのエラ

一によって、レールとしての同じゾーン内のCPUが異なった情報を処理しないことを保証することである。もしこのようなことが発生すれば、CPUモジュール30は故障を検出し、これによってドラチックだが恐らく必要のないアクションが発生する。このことを回避するため、クロスリンク90はEXCLUSIVE ORゲート960を有し、このゲートによってマルチプレクサ945と945mの出力が比較される。もしこれらの出力が異なっていれば、ゲート960によってマルチプレクサ949はERRコードを選択する。EXCLUSIVE ORゲート960mは、同様にマルチプレクサ949mにまたERRコードを選択させる。このコードは、エラーが発生しているがCPUモジュールにエラーの発生することは回避されていることをメモリ制御装置70と75に示す。メモリ・モジュール60に対するシングル・レール・インターフェースはデータとアドレスに対して同じ結果を達成する。

第12図に示すデータとアドレスの流れは第

マルチプレクサMUXD982に対する入力を与えられる。マルチプレクサMUXD982の他方の入力はラッチ980の出力であり、このラッチ988はラッチ978から入力されたデータとアドレスを有している。マルチプレクサ982は次にその入力の1つを選択し、こらは次にラッチ900に記憶される。もしデータまたはアドレスがメモリ制御装置70に送られるべきであれば、ドライバ922が起動される。シリアル・レジスタ920からのデータはドライバ944を介してメモリ制御装置70に送られる。

クロスリンク90を通るデータ、特に第11図および第12図の両方のエクソネオール(xonreol)素子を通るデータは、デコード・ロジック970、デコード・ロジック971、デコードロジック996、およびデコード・ロジック998によって発生される幾つかの信号によって制御される。適当な入力ソースを選択するため、このロジックによって、適当な入力ソースを選択するために、マルチプレクサ935、942、945、949、

11図の制御信号の流れと同じである。モジュール相互接続部130からのデータとアドレスは、ラッチ972に記憶され、次に入力としてマルチプレクサMUXB974に入力として加えられる。並列レジスタ910からのデータによって別の入力がマルチプレクサ974に加えられる。マルチプレクサ974の出力は、マルチプレクサMUXC976に対する入力であり、このマルチプレクサMUXC976は、またもともとメモリ制御装置70から送られてラッチ961に記憶されているデータとアドレスを受け取る。マルチプレクサ976は、次にこれらの入力の1つを選択してラッチ798に記憶する。もしモジュール相互接続部130から入力されたものであれば、メモリ制御装置70から入力されたものであれば、もしデータとアドレスがクロスリンク90'に送られるべきであれば、ドライバ984が起動される。

クロスリンク90'から入力されたデータは受信装置986によってバッファされラッチ988に記憶されるが、このラッチ988によってまた

966、974、976、および982を制御する信号を与えられる。更に、このデコード・ロジックは、またドライバ940、946、951、969、984、992、および994を制御する。

制御信号の大部分は、デコード・ロジック998によって発生されるが、これらの一部はデコード・ロジック970、971、970m、971m、および996によって発生される。デコード・ロジック998、970および970mは、データとコードがそれ自身のゾーンから受け取られるか他のゾーンから受け取られるかを制御するのに必要なデータとコードをこのロジックが受け取ることを保証する位置に持続される。

デコード・ロジック971、971mおよび966の目的は、ドライバ937、937mおよび984が適切な状態にセットされることを保証することである。この「初期デコード」によって、データ・アドレスとコードが全てのケースで適切なクロスリンクに送られることを確認する。この

ような初期デコード・ロジックがなければ、クロスリンクは全てそれらのドライバが不能にされた状態におかれる可能性がある。メモリ制御装置のドライバがまた不能にされれば、そのクロスリンクは決してアドレス、データおよび制御コードを受け取らず、そのクロスリンクに接続されているI/Oモジュールの全てを効率的に不能にする。

デコード・ロジック970、971、970m、971m、および998によって発生されたドライバ制御信号を説明する前に、これらのゾーン、従ってクロスリンク90と95がとることのできる異なったモードを理解する必要がある。第13図は、異なった状態AないしFおよび各モードに対応するこれらの状態を説明する表である。

開始時およびその他の場合、両方のゾーンは状態Aにあり、この状態Aはこれら両方のゾーンに対するOFFモードとして知られる。このモードの場合、両方のゾーンのコンピュータ・システムは独立して動作している。これらのゾーンの1つの動作システムが他方のゾーンのI/Oと通信を

行う能力を要求し、その要求が受け入れられた後、これらのゾーンは状態BとCとして示されるマスター/スレーブ・モードに入る。このようなモードの場合、マスターであるゾーンは動作しているCPUを有し、そのゾーンおよび他方のゾーンのI/Oモジュールを制御する。

再同期を開始すると、コンピュータ・システムは状態BまたはCのいずれかのマスター/スレーブモードを離脱し、状態EおよびFとして示される再同期スレーブ/再同期マスター・モードに入る。これらのモードの場合、マスター・ゾーンであったゾーンが他方のゾーンのCPUをオン・ラインにする役割を果たす。もし再同期に失敗すれば、これらのゾーンは前に再同期しようとしたのと同じマスター/スレーブモードに戻る。

しかし、もし再同期が成功すれば、これらのゾーンは状態Dに入り、この状態Dは完全デュプレックス・モードである。このモードの場合、両方のモードはロックステップ同期状態で共に動作する。動作は、CPM/MEMの故障が発生する迄、

このモードで継続され、この場合、システムは2つのマスター・スレーブ・モードの1つに入る。スレーブはそのプロセッサがCPM/MEM故障を経験したゾーンである。

状態D、すなわち完全デュプレックス・モードで作動している場合、最も顕著なのはクロック位相エラーであるが、ある種のエラーが発生すると、システムを2つの独立した処理システムに分割する必要が生ずる。これによってシステムは状態Aに戻る。

第11図および第12図に示すデコード・ロジック970、970m、971、971m、998（まとめてクロスリンク・制御ロジックと称する）は、クロスリンク・ドライバとマルチプレксаをどのようにして適切な状態にセットするかを決定するため、第10図に示す再同期モード・ビット915とクロスリンク・モード・ビット916にアクセスする、更に、このクロスリンク・デコード・ロジックは、またデータ・トランザクションの期間中にメモリ制御装置70と75から送られ

たアドレスの一部を受け取って分析し、クロスリンク・マルチプレксаとドライバの状態をどのようにして設定するかをクロスリンク・デコード・ロジックに対して更に指示すアドレス情報を取り出す。

マルチプレксаの状態を設定するのに必要な情報は、一度異なったモードとトランザクションを理解すると、かなりははっきりする。行うべき唯一の判断はデータのソースである。従って、クロスリンク90と95がスレーブ・モードにある場合、マルチプレкса935、935m、および966はゾーン11からデータ・アドレスとコードを選択する。もしクロスリンク90と95が完全にデュプレックス・モードにあり、I/Oの命令のアドレスがゾーン11のI/Oに接続された装置に対するものであり、影響を受けたマルチプレксаとのクロスリンクがクロスオーバー・モードにあれば、これらのマルチプレксаはまた他方のゾーンからデータ、アドレスおよびコードを選択する。クロスオーバー・モードの場合、モジュール相互

接続部に送られるべきデータはチェックのため他方のゾーンから受け取られるべきである。好適な実施例の場合、モジュール相互接続部130はゾーン11のプライマリ・レールからデータ、アドレスおよびコードを受け取り、モジュール接続部は、ゾーン11'のミラー・レールからデータ、アドレスおよびコードを受け取る。または、モジュール相互接続部132はゾーン11'のプライマリ・レールからデータ、アドレスおよびコードを受け取ることができ、これによって、一方のゾーンのプライマリ・レールを他方のゾーンのミラー・レールと比較することが可能になる。

マルチプレクサ945、945m、982は、データのソースであるいずれかのゾーンからデータ、アドレスおよびコードを受け入れるようにセットされる。このことは、全てのクロスリンクが完全にデュプレックス・モードにあり、データ、アドレスおよびコードがI/Oモジュールから受け取られる場合と、クロスリンクが再同期スレーブ・モードであり、データ、アドレスおよびコー

ドが他方のゾーンのメモリ制御装置から受け取られる場合の両方について、真実である。

もしメモリ制御装置70および75からのアドレス情報が、応答データとコードのソースがクロスリンク自身の並列レジスタ910であることを示せば、マルチプレクサ942、942m、および974はこれらのレジスタからデータとコードを選択するようにセットされる。同様に、もしメモリ制御装置70および75からのアドレス情報が応答データのソースはクロスリンク自身のシリアル・レジスタ920であることを示せば、マルチプレクサ949と949mはデータとコードをこれらのレジスタから選択するようにセットされる。

もしこの情報がメモリ再同期動作期間中の制御コードであれば、マルチプレクサ949と949mはデコード・ロジック970と970mからデータを選択するようにまたセットされ、もしEXCLUSIVE ORゲート960と960mがクロスリンク90と95を介して転送された

データの間で比較のミスを識別すれば、ERRORコードを選択するようにセットされる。この後者の場合、マルチプレクサ949と949mの制御は、クロスリンクロジックからではなくてEXCLUSIVE ORゲート960と960mから行われる。マルチプレクサ949と949mは、クロスリンク・レジスタ910が要求された場合には、これらのレジスタからコードをまた選択し、これらのコードが要求された場合には、マルチプレクサ945と945mの出力をまた選択する。マルチプレクサ945と945mは、それぞれマルチプレクサ942と942mからの出力かまたはそれぞれクロスリンク90'と95'からのI/Oコードかのいずれかを選択する。

マルチプレクサ976は、I/Oモジュールとのトランザクションの場合には、モジュール相互接続部139からデータとアドレスを選択するか、またはデータとアドレスがI/Oに対してかまたはメモリの再同期の期間中かのいずれかにクロスリンク90'に送られるべきである場合、メモリ

制御装置90からのデータとアドレスを選択するかのいずれかである。

ドライバ937と937mは、クロスリンク90と95がデュプレックス・モード、マスター・モードまたは再同期マスター・モードにある場合、動作される。ドライバ940と940mは、ゾーン11のI/Oトランザクションの場合に動作される。ドライバ946と946mは、クロスリンク90と95がデュプレックス・モードまたはスレーブ・モードの場合に動作される。ドライバ951と951mは常に動作されている。

ドライバ969はゾーン11に対するI/O書き込み期間中に動作される。ドライバ984は、クロスリンク90がデータとアドレスをゾーン11'のI/Oに送っている場合、またはクロスリンク90が再同期マスター・モードにある場合に動作される。受信機986はクロスリンク90'からデータを受け取る。ドライバ992と994は、データがメモリ制御装置70に送られている場合に動作される。ドライバ994は、シリアル

・クロスリンク・レジスタ910の内容が読み出されている場合に動作され、ドライバ992は全ての他の読み出し期間中に動作される。

5. 発振器

両方の処理システム20と20'が各々同じ機能を完全デュープレックス・モードで実行している場合、CPUモジュール30と30'が同じ速度で動作を実行することが避けられない、もしそうでなければ、処理時間の大部分は、1'0およびインタープロセッサのエラーのチェックのために処理システムの20と20'を再同期させることに消費されてしまう。処理システム20と20'の好適な実施例の場合、これらのシステムの基本的なクロック信号は相互に同期されて位相ロックされている。故障許容コンピュータ・システム10は、処理システム20と20'に対するクロック信号の周波数を制御し、各処理システムに対するクロック信号の間の位相差を最小にするために、タイミング・システムを有している。

第14図は、処理システム20と20'で実施

される本発明のタイミング・システムのブロック図を示す。このタイミング・システムは、処理システム20のCPUモジュール30の発信器システム200と処理システム20'のCPUモジュール30'の発振器システム200'によって構成される。発振器200'の構成要素は発振器200の構成要素と同じであり、両方の発振器システムの動作は同じである。従って、発振器システム200と200'の動作が異なっている場合を除いて、発振器システム200の構成要素と動作のみを説明する。

第14図に示すように、発振器システム200の大部分、特にデジタル・ロジックはクロスリンク95内部に位置しているが、この位置は本発明にとって必要なものではない。発振器システム200は電圧制御水晶発振器(VCX)205を有し、これは好ましくは、66.66 Mhzの基本発振器信号を発生する。VCX205の周波数は入力電圧レベルによって調整することができる。クロック分配チップ210は基本発振器信号を

分周し、全て同じ周波数を有する4つの一次クロックを発生することが望ましい。プライマリCPU40の場合、これらのクロックはPCLKLおよびPCLKHであり、これらは相互に論理が反転しているものである。ミラーCPU50の場合、クロック分配チップ210はクロック信号MCLKLとMCLKHを発生し、これらはまた相互に論理が反転しているものである。第15図は、これらのクロック信号のタイミングと位相の関係を示す。クロック信号PCLKL、PCLKH、MCLKM、およびMCLKHは約33.33 Mhzであることが望ましい。クロック・チップ210は、また第15図に示す16.66 Mhzの位相ロック・ループ信号CLKCHをまた発生する。この位相ロック・ループ信号は、この信号をバッファするクロック・ロジック220に送られる。

クロック・ロジック・バッファ220は、同期に使用するため、CLKCH信号を発振器200'に送る。発振器200'のクロック・ロジック・

バッファ220'は、それ自身のバッファされた位相ロック・ループ信号CLKCHを発振器200の位相検出器230に送る。位相検出器230は、遅延素子225を介してクロック・ロジック220から位相ロック・ループ信号CLKCHをまた受け取る。遅延素子225は、クロック・ロジック・バッファ220'からのケーブル・ラン(cable run)による遅延を概算する。

位相検出器230は、その入力位相ロック・ループ信号を比較して2つの出力を発生する。これらの信号の1つは位相差異信号235であり、これはループ増幅器240を介してVCX205の電圧入力に送られる。位相差異信号235によって、増幅器240は信号を発生し、この位相差異を補償するためにVCX205の周波数を変換する。

位相検出器230の他方の出力は、位相エラー信号236であり、これは可能性のある同期の故障を示す。

第16図は、位相検出器230の詳細図である。

位相検出器230は位相比較器232と電圧比較器234を有する。位相比較器232は、遅延素子225からクロック信号(CLKCH)を受け取ると共に検出器200'から位相ロック・ループ・クロック信号(CLKCH)を受け取り、これらの信号の位相差を表す電圧差として位相差信号235を発生する。

もしクロックを同期させる目的のために処理システム20が「スレーブ」であれば、スイッチ245は「SLAVE」の位置(すなわち閉)にあり、電圧水準235は、ループ増幅器240によって増幅された後、VCXO205の周波数を制御する。もし両方のスイッチ245と245'が「マスター」の位置にあれば、処理システム20と20'は位相ロックされず、非同期の状態(独立して)動作する。

位相差信号235の電圧水準は、また電圧比較器234に対する入力であり、これらの位相差は位相の進みと遅れの許容範囲を表す電圧 V_{ref1} および V_{ref2} である。もしこの位相差が許容範囲で

あれば、PHASE ERROR信号は活性化されない。もしこの位相差が許容範囲以外であれば、PHASE ERROR信号236は活性化され、クロック・デコーダ220を介してクロスリンク95に送られる。

6. I/Oモジュール

第17図はI/Oモジュール100の好適な実施例を示す。このI/Oモジュール100の動作の原理は、他のI/Oモジュールにも同様に適用することができる。

第18図はファイヤウォール(firewall)1000の好適な実施例の構成要素を示す。ファイヤウォール1000は、第17図に示すモジュール相互接続部130に対する16ビットのバス・インターフェース1810とバス1020に接続するための32ビットのバス・インターフェース1820を有する。インターフェース1810と1820は内部ファイヤウォール・バス1815によって接続され、このファイヤウォール・バス1815はまたファイヤウォール1000の他の構成要素と

も相互に接続される。バス1815は16または35ビット幅の並列バスであることが望ましい。

I/Oモジュール100はデュアル・レール・モジュール相互接続部130と132によってCPUモジュール30に接続される。モジュール相互接続部の各々は、それぞれファイヤウォール1000と1010に接続される。通常はファイヤウォール1000であるが必ずしもこれではない一方のファイヤウォールは、モジュール相互接続部130からバス1020にデータを書き込む。この場合にはファイヤウォール1010である他方のファイヤウォールは、第18図に示すファイヤウォール比較回路1840を使用して、そのデータをモジュール相互接続部132から受け取った自分自身のコピーとチェックする。このチェックは有効であるが、その理由は、CPUモジュール30と30'からI/Oモジュールに対して書き込まれたデータを実質的に同時にファイヤウォール1000と1010で入手可能にしているこれらのCPUモジュール30と30'がロックス

テップ同期の状態にあるからである。

ファイヤウォール比較回路1840は、CPUモジュール30と30'から受取ったデータのみをチェックするだけである。I/O装置からCPUモジュール30と30'送られたデータは、共通の供給元を有し、従ってチェックを必要としない。その代わり、I/O装置から受取られCPUモジュール30と30'に送られるデータは、EDC/CRC発生装置1850によって実行される周期的冗長性チェック(CRC)コードのようなエラー検出コード(EDC)によってチェックされる。EDC/CRC発生装置1850は、また内部ファイヤウォール・バス1815に接続される。

EDC/CRC発生装置1850は、I/O装置によって使用されるのと同じEDC/CRCコードを発生してチェックを行う。I/Oモジュール100は2つのEDCを発生することが望ましい。一方のEDCはまたEDC/CRCでもよく、これはモジュール100が接続されているアサネット(Ethernet)パケット・ネットワークのよ

うなネットワークに対するインターフェースに使用される(第17図の構成要素108に参照)。他方のEDCは第17図のディスク・インターフェース1072のようなディスク・インターフェースに使用される。

CPUモジュール30とI/Oモジュール100との間でEDC/CRCを適応することは必要でないが、その理由は、モジュール相互接続部が2重になっているからである。例えばCPUモジュール30の場合、クロスリンク90はモジュール相互接続部30を介してファイヤウォール1000と通信を行い、クロスリンク95はモジュール相互接続部132を介してファイヤウォール1000と通信を行う。

アサネット・ネットワーク1082から受け取られたメッセージは、第17図に示すネットワーク制御装置1080によってEDC/CRCの有効性をチェックされる。EDC/CRCが完全であるデータは、これもまた第17図に示すローカルRAM1060に書き込まれる。ローカル

RAM1060内の全てのデータは、DMAを使用してメモリ・モジュール60に転送される。

DMA制御装置1890は転送の調整を行い、EDC/CRC発生装置に転送中のEDC/CRCによって符号化されたデータの有効性をチェックさせる。

I/O装置との大部分のデータの転送はDMAによって行われる。データはメイン・メモリとI/Oバッファ・メモリとの間を移動する。データがメイン・メモリからI/Oバッファメモリに移動する場合、EDC/CRCを付加してもよい。データがI/Oバッファメモリからメイン・メモリに移動する場合、EDC/CRCはチェックを受けてメイン・メモリに移動してもよく、または取り除かれてもよい。データがI/Oバッファメモリからディスクまたはアサネット・アダプタのような外部装置を介して移動される場合、EDC/CRCは局部的または離れた位置にある受信ノードでチェックされてもよく、またはその両方でチェックされてもよい。メモリ・データ・パケッ

トは遠くの位置にあるノードまたはI/Oモジュールのローカル・インターフェースによって発生されたそれらのEDC/CRCを有してもよい。

この動作によって、I/Oモジュール100のようなシングル・レール・システムに存在する、またはこれを介して転送中のデータがエラー検出コードによってカバーされることが保証され、このエラー検出コードはこのデータが最終的に通過する通信メディアと少なくとも同じくらい信頼性のあることが望ましい。例えば、同期プロトコルを処理するような異なったI/Oモジュールは、適当なプロトコルのEDC/CRCコードを発生してチェックするEDC/CRC発生装置を有することが望ましい。

一般的に、DMA制御装置1890はアドレスされている共有のメモリ制御装置105とローカルRAM1060に特有のDMAの動作の部分を取扱う。32ビット・バス1020は2つの異なったモードで駆動される。DMAのセットアップの期間中、DMA制御装置1890は標準非同期

マイクロプロセッサ・バスとしてバス1020を使用する。DMAの動作が発生するローカルRAM1060のアドレスは共有のメモリ制御装置105とDMA制御装置1890に供給される。実際のDMAの転送の期間中、DMA制御装置1890はDMA制御線1895に非同期的状態でバス1020を駆動させる。共有のメモリ制御装置1050はバス・サイクル毎に32ビットのデータ・ワードをバス1020に転送し、DMA制御装置1090はどれくらいの数のワードの転送が残っているかについての情報を得る。共有のメモリ制御装置1050は、またローカルRAM1060を制御して次のDMAアドレスを発生する。

I/Oモジュール(100、110、120)はそれら自身のローカルRAM1060に対する読み出し/書き込み動作を制御する責任を負う。

CPUモジュール30はメモリ・アレイ60との転送動作を制御する責任を負う。メモリ制御装置70と75のDMAエンジン800(第8図に示す)は、CPUモジュール30に対するDMAの

動作を管理する。このような作業の分割によって、いずれかのモジュールのDMAロジックの故障がゾーン11または11'のいずれかの他のモジュールのデータの健全性を低下させることを防止する。

トレースRAM1872はトレースRAM制御装置1870の機能を以下で詳細に説明する。簡単に言えば、故障が検出され、CPU40、40'、50および50'とCPUモジュール30および30'がそのことを通知されると、コンピュータ・システム10全体の種々のトレースRAMが以下で説明するある種の機能を実行する。トレースRAMとの通信はトレース・バス1095で行われる。トレースRAM制御装置1870は、トレース・バス1095からの信号に応答して、トレースRAM1872に記憶を停止させるかその内容をソレース・バス1095放出させる。

32ビットの並列バスであることが望ましいI/Oモジュール・バス1020は、ファイヤウォール1000および1010に接続されると共に

I/Oモジュール100の他の構成要素にも接続される。共有のメモリ制御装置1050は、I/Oモジュール100のI/Oモジュール・バス1020にもまた接続される。共有のメモリ制御装置1050は共有のメモリ・バス1065によってローカル・メモリ1060に接続され、この共有のメモリ・バス1065は32ビットのデータを搬送することが望ましい。ローカル・メモリ1060は256キロバイトのメモリを有するRAMであることが望ましいが、このRAM1060は任意のサイズでよい。共有のメモリ制御装置1050とローカルRAM1060によって、I/Oモジュール100に対する記憶能力が与えられる。

ディスク制御装置1070によって、第1図のディスク1075および1075'のようなディスクに対して標準のインターフェースが設けられる。ディスク制御装置1070は、ローカルRAM1060に使用するためまたはI/Oモジュール・バス1020との通信を行うために共有のメモ

リ制御装置1050にまた接続される。

ネットワーク制御装置1080はネットワーク・インターフェース1082によってETHERNETネットワークのような標準ネットワークに対してインターフェースを与える。ネットワーク制御装置1080は、ローカルRAM1060とI/Oモジュール・バス1020の両方に対してインターフェースとして機能する共有のメモリ制御装置1050にまた接続される。しかし、I/Oモジュール・バス1020の特定の組織または構造については何等の要求も存在しない。

PCIM（電源および冷却用インターフェース・モジュール）サポート・エレメント1030は、I/Oモジュール・バス1020に接続されると共にASCIIインターフェース103に接続される。PCIMサポート・エレメント1030によって、処理システム20は電源システムの状態（すなわちバッテリー・レギュレータ等）と冷却システム（すなわちファン）を監視してこれらの適切な動作を保証することが可能になる。PCIM

サポート・エレメント1030は、バッテリーの電圧が許容できない程度に低い等のある種の故障または潜在的な故障の徴候が存在する場合のみ、メッセージを受け取ることが望ましい。全ての電源および冷却サブシステムを周期的に監視するために、CPIMサポート・エレメント1030を使用することもまた可能である。または、PCIMサポート・エレメント1030は、直接ファイヤウォール1000と1010に接続されてもよい。

診断マイクロプロセッサ1100が、またI/Oモジュール・バス1020に接続される。一般的に、診断マイクロプロセッサ1100は、故障が検出された場合、トレースRAM1872のようなトレースRAMからエラー・チェック情報を集めるために使用される。このデータは、それぞれファイヤウォール1000と1010を介してトレース・バス1095と1096に集められると共にモジュール・バス1020を介してマイクロプロセッサ1100に集められる。

D. インタープロセッサとインターモジュールの

通信

1. データ経路

コンピュータ・システム10の構成要素は、それら自身によって故障許容システムを構成するものではない。正常な動作の期間中および故障の検出と修正の動作の期間中に通信を可能にする通信経路とプロトコルが必要である。このような通信信号に対するキーは、クロスリンク経路25である。クロスリンク経路25は、並列リンク、直列リンク、および既に説明したクロック信号によって構成される。これらは19図に示される。並列リンクは、2組の同じデータおよびアドレス線、制御線、割り込み線、符号化エラー線、および1本のソフト・リセット・リクエスト線を有する。データおよびアドレス線と制御線は、モジュール相互接続部130と132（または130'と132'）またはメモリ・モジュール60（60'）からCPUモジュールの間で交換される情報を有している。

割り込み線は、I/Oサブシステム（モジュール

関係なくデュアル・レール・システムとして動作を継続するように設計されている。I/Oサブシステム（モジュール100、110、120、100'、110'、120'）は、また過渡的なエラーまたは故障を経験しても動作を継続することができる。好適な実施例の場合、ファイナル比較回路1840の検出したエラーによって、同期化されたエラー・レポートがCPUの管理する動作に関して経路25を介して行われる。CPU30と30'のハードウェアは経路25を介して同期化されたソフト・リセットを行い、故障のある動作をもう一度行う。DMAの管理する動作の場合、同じエラーの検出によって、同期割り込みが経路235を介して行われ、CPU40、50、40'、および50'のソフトウェアはDMAの動作を再び開始する。

ある種の過渡的なエラーは、動作を完全デブレックスの同期形態で継続するように直ちに修復されるものではない、例えば、メモリ・モジュール60に制御エラーが発生すると、その結果メモ

リ100、110、120、100'、110'および120'）で使用可能な割り込み水準の各々に対し1本の線を有することが望ましい。これらの線はクロスリンク90、95、90'、および95'によって共有される。

符号化エラー線は、両方のゾーンに対するコンソール「HALT」要求を同期させる複数のコードを有することが望ましく、これらの複数のコードの1つは両方のゾーンに対してCPUエラーを同期させるコード、1つは他方のゾーンに対してCPU/メモリの故障の発生を示すコード、1つは両方のゾーンに対してDMAエラーを同期させるコード、および1つはクロック位相エラーを示すコードである。各ゾーン11または11'からのエラー線は、ゾーン11に対するORゲート1990またはゾーン11'に対するORゲート1990'のようなORゲートに対する入力である。各ORゲート2の出力によって、他方のゾーンのクロスリンクに対する入力を与えられる。

欠陥許容処理システム10は、過渡的な故障に

り、モジュール60に未知のデータが生じる。この場合、CPUとメモリ・エレメントは最早フェール・セーフ・システムの一部として信頼性のある機能は果たすことはできず、従ってこれらを取り外さなければならない。メモリ・アレー60はそこで、CPUとメモリ・エレメントが再びメモリに取り付けられる前に、メモリの再同期を行わなければならない。経路25の符号化エラー線のCPUメモリ故障コードは、CPU30のCPUとメモリ・エレメントが故障していることをCPU30'に知らせる。

サイクル・タイプ、エラー・タイプおよび準備完了状態の組み合わせを示す制御線によって、CPUモジュール（30および30'）とI/Oモジュールとの間にハンドシェーキングが行われる。上で説明したように、実行されているバス動作のタイプがサイクル・タイプによって決められる。すなわち、これらは、CPU I/Oの読み出し、DMAの転送、DMAのセットアップまたは割り込みベクトルの要求である。エラー・タイ

ブによってファイウォールの比較ミスまたはCRCのエラーが決められる。「準備完了」のメッセージはCPUとI/Oモジュールとの間に送られて要求された動作の完了を示す。

シリアル・クロスリンクは状態読み出しのためのシリアル・データの転送、ループバック、およびデータの転送を行うために2本の線を2組有している。

交換されるクロック信号は、位相ロック・クロック信号CLKC HとCLKK' H(遅延した)。である。

第20A図乃至第20D図は、異なった動作期間中にデータが通過するCPUモジュール30および30'とI/Oモジュール100および100'の構成要素のブロックを示す。これらの構成要素の各々は前に説明したものである。

第20A図は、共有のメモリ制御装置1050(1050')からのレジスタ・データのCPUのI/Oレジスタによる読み出し動作のようなI/Oモジュール100からのデータの一般的な

CPU I/Oによるデータ読み出し動作のためのデータ経路を示す。このような動作はローカル・データの読み出しと呼び、これをローカル・メモリ1060からのDMAによるデータの読み出しと区別し、このローカル・メモリ1060は通常内部装置の制御装置からのデータを有している。ローカル・データは共有のメモリ制御装置1050(1050')を介して転送されるようにローカルRAM1060(1060')に記憶されているものと仮定する。経路が1つの場合、データはファイアウォール1000、モジュール相互接続部130を介してクロスリンク90に流れる。第12図から分かるように、クロスリンク90はファイアウォール1000からメモリ制御装置90に流れるデータを遅延させ、その結果、クロスリンク90'に対するデータは、データがメモリ制御装置70に加えられるのと同時に、このメモリ制御装置70に加えられ、従って、処理システム20と20'が同期状態のままであることが可能になる。このデータは、次に内部バス46と

40'によってメモリ制御装置70および70'からCPU40および40'に進む。

同じ経路を使用してCPU50と50'にデータを読み込む。共有のメモリ制御装置1050からのデータはファイアウォール1010を介してクロスリンク95に進む。この時、データはクロスリンク95'と遅延装置を介してクロスリンク95の内部の両方に流れる。

CPU I/O読み出し動作は、また共有のメモリ制御装置1050'とI/O装置100'のローカルRAMを介して処理システム20'のI/O処置から受け取られたデータに対してもまた実行されることができる。

I/Oモジュール100、110、および120は同じものであり、それぞれI/Oモジュール100'、110'、120'に対応するが、対応するI/Oモジュールはロックステップ同期状態にはない。CPU I/O読み出しのためメモリ制御装置1050'とローカルRAM1060'を行う使用して、データは先ずクロスリンク90'

と95'に進む。残りのデータ経路はメモリ制御装置1050からの経路と同じである。データはクロスリンク90'と95'からメモリ制御装置70'と75'を経由して最終的にそれぞれCPU40'と50'に進む。同時に、データはそれぞれクロスリンク90と95を横切って進み、次に遅延エレメントを経由しないでそれぞれCPU40と50に進み続ける。

第20B図は、ローカル・データのCPU I/O書き込み動作を示す。このようなローカル・データはCPU40、50、40'および50'からI/Oモジュール100のようなI/Oモジュールに転送される。このような動作の1つの例は、共有のメモリ制御装置1050におけるレジスタAに対する書き込みである。CPU40によって転送されるデータは同じ経路に沿って進むが、その方向はCPU I/Oの読み出し期間中のデータの方向と逆の方向である。特に、このようなデータはバス46、メモリ制御装置70、種々のラッチ(同期を行うため)、ファイアウォール

1000、およびメモリ制御装置1050を通過する。CPU50'からのデータは、またCPU1/Oの読み出しの経路を逆の方向に流れる。特に、このようなデータは、バス56'、メモリ制御装置75'クロスリンク95'クロスリンク95を経由し(ファイヤウォール1010に行く。上で述べたように、ファイヤウォール1000と1010はI/Oの書き込み動作の期間中にデータをチェックして記憶する前にエラーを調べる。

書き込みが他方のゾーンのI/Oモジュールに対して行われる場合、同じ動作が行われる。しかし、CPU50と40'からのデータがCPU50'と40からのデータの代わりに使用される。

CPU50と40'からのデータは対称の経路を介して共有のメモリ制御装置1050'に転送される。CPU50と40'からのデータはファイヤウォール1000'と1010'によって比較される。I/O書き込みデータに対してサービスを行うために異なったCPUの対が使用される理由は、完全デュプレックス・システムで正常に

使用している期間中に全てのデータ経路をチェックするためである。各ゾーンに対するインターレース・チェックはメモリ制御装置70、75、70'および75'で前に実行された。

第20C図は、DMA読取り動作に対するデータ経路を示す。メモリ・アレイ600からのデータは、同時にメモリ制御装置70と75に入り、次いでクロスリンク90と95に入る。クロスリンク90はファイヤウォール1000に転送されたデータを遅延させ、その結果、クロスリンク90と95'からのデータは実質的に同じ時間にファイヤウォール1000と1010に到着する。

CPU I/O書き込み動作と同様に、種々のクロスリンクに対するデータの4つのデータ/コピーが存在する。ファイヤウォールでは2つのコピーのみが受け取られる。ゾーン11に対する読み出しを実行する場合には、異なった対のデータが使用される。DMAの書き込み動作に対するデータ経路は第20D図に示され、これらはCPU I/Oの読み出しに対するデータと同じである。

特に、共有のメモリ制御装置1050'からのデータは、ファイアウォール1000'、クロスリンク90'(遅延を伴う)、メモリ制御装置70'を経由してメモリ・アレイ600'に進む。同時に、このデータは、ファイヤウォール1010'クロスリンク95'(遅延を伴う)およびメモリ制御装置75'を通過し、この時これはインターレース・エラー・チェックの期間中にメモリ制御装置70'からのデータと比較される。CPU I/Oの読み出しの場合のように、DMA書き込み動作中のデータは、共有のメモリ制御装置1050を介して交互に同じ動作に入ってもよい。

クロスリンク90'からのデータは、またクロスリンク90とメモリ制御装置70を通過してメモリ・アレイ600に行く。クロスリンク95'からのデータは、クロスリンク95とメモリ制御装置75を通過し、この時これは同時に行われるインターレース・チェックの期間中にメモリ制御装置70'からのデータと比較される。

第20E図は、メモリ再同期(resync)動作の

ためのデータ経路を示す。この動作の場合、メモリ・アレイ60と60'の両方の内容は、相互に同じように設定されなければならない。メモリの再同期の場合、メモリ・アレイ600'からのデータは、DMAに制御されてメモリ制御装置70'と75'を通過し、次にそれぞれクロスリンク90'と95'を通過する。このデータは、次にメモリ600アレイに記憶される前に、それぞれメモリ制御装置70と75に入る。

2. リセット

システム10に関する上記の議論は、リセットに関する多くの異なった必要性を考慮して行われた。議論しなかったある種の場合には、リセットは、電源が最初にシステム10に印加される場合等の標準的な機能のために行われる。多くのシステムは1つのリセットを有し、このリセットは常にプロセッサをある所定の状態または最初の状態にセットし、従ってプロセッサの命令の流れを中断する。しかし、大部分の他のシステムと異なって、システム10のリセットは、もし絶対的に必

要でなければ、CPU 40、40'、50および50'による命令の実行の流れに影響を及ぼさない。更に、システム10のリセットは、正常な動作を回復するためにリセットされる必要のある部分のみに影響を及ぼす。

システム10のリセットの他の特徴は、これらのリセットの抑制である。故障許容システムの最も重要な考慮すべき事項の1つは、もしある機能が故障しても、その機能はシステムの動作を停止してはならないことである。この理由のため、システムのいかなる1つのリセットも、ゾーン11と11'が直接に協力しないなら、ゾーン11と11'の両方の構成要素を制御することはできない。従って、完全デブレックス・モードで動作している場合、ゾーン11内の全てのリセットはゾーン11'内のリセットとは独立している。しかし、システム10がマスター／スレーブ・モードにある場合、スレーブゾーンはマスターゾーンのリセットを使用する。更に、システム10内のいかなるリセットもメモリ・チップの内容に影響

を及ぼさない。従って、キャッシュ・メモリ42及び52、スクラッチ・パッド・メモリ45および55またはメモリ・モジュール60のいずれもリセットによっていかなるデータも失うことはない。

システム12は3つのクラスのリセット、すなわち、「クロック・リセット」「ハード・リセット」、および「ソフト・リセット」があることが望ましい。クロック・リセットはゾーン内の全てのクロック位相発生器を再編成する。ゾーン11内のクロック・リセットはまたCPU 40と50、およびメモリ・モジュール60をイニシャライズする。クロック・リセットは、これらのモジュールのクロック位相発生器を再編成する以外にモジュール相互接続部130と132に影響を及ぼさない。システム10がマスター／スレーブモードにある場合でさえ、スレーブ・ゾーンでクロック・リセットを行っても、これはマスターゾーンのモジュール相互接続部からスレーブ・ゾーンのモジュール相互接続部に対するデータの転送を妨げ

ない。しかし、ゾーン11'でクロック・リセットを行うと、ゾーン11'内の対応する構成要素がイニシャライズされる。

一般的に、ハード・リセットを行うと、全ての状態デバイスとレジスタはある所定の状態または最初の状態に戻る。ソフト・リセットを行うと、状態エンジンと一時的に記憶を行うレジスタのみがそれらの所定の状態または最初の状態に戻るだけである。1つのモジュール内の状態エンジンはそのモジュールの状態を決める回路である。エラー情報と構成データを有するレジスタはソフト・リセットによって影響を与えられない。更に、システム10は、処理を継続するために、再びイニシャライズされる必要のある構成要素のみをリセットするために同時にハード・リセットとソフト・リセットの両方を選択的に行う。

ハード・リセットはシステム10をクリアし、従来のシステムと同様に、システム10を既知の構成に戻す。ハード・リセットは、ゾーンが同期されるべき場合または1/Oモジュールをイニシ

ャライズまたは不能にするべき場合に、電源を印加した後、使用される。システム10の場合、4つのハード・リセット、すなわち、「パワーアップ・リセット」、「CPUハード・リセット」、「モジュール・リセット」、及び「デバイス・リセット」があることが望ましい。ハード・リセットは更にローカル・ハード・リセットとシステム・ハード・リセットに分けることができる。ローカル・ハード・リセットは、CPUがスレーブ・モードにある場合に応答するロジックのみにに影響を及ぼす。システム・ハード・リセットは、クロスリンク・ケーブル25とモジュール相互接続部130及び132に接続されているロジックのみに限定される。

パワーアップ・リセットは、電源が印加された直後に、ゾーン11と11'をイニシャライズするために使用される。パワーアップ・リセットによって、ゾーンの全ての部分に対して強制的にリセットが行われる。パワーアップ・リセットはシステム11のゾーンの間では決して接続されない

が、その理由は、各ゾーンがそれ自身の電源を有し、従って異なった長さの「電源投入」イベントを経験するからである。パワーアップ・リセットは全てのハード・リセットとクロック・リセットをゾーン11または11'に行うことによって実行される。

CPUハード・リセットは、CPUモジュールを既知の状態に戻すため診断目的に使用される。CPUハード・リセットは影響の与えられたゾーン内にあるCPU、メモリ制御装置、およびメモリ・モジュール、状態レジスタの全ての情報をクリアする。キャッシュ・メモリとメモリ・モジュールは不能にされるが、スクラッチ・パッドRAM 45および55の内容とメモリ・モジュール60の内容は変化されない。更に、パワーアップ・リセットと違って、CPUハード・リセットはクロスリンクのゾーン識別またはクロック・マスタシップを変更しない。CPUハード・リセットは、CPUモジュールとクロック・リセットに加えることのできる全てのローカル・ハード・リセット

の合計である。

・モジュール・ハード・リセットは、ルートストラッピングの期間中のような既知の状態にI/Oモジュールをセットするために使用され、また故障したI/Oモジュールをシステムから取り外すためにも使用される。I/Oモジュール・ハイド・リセットはモジュール上の全てのものをクリアし、診断モードでファイアウォールを離れ、ドライバを不能にする。

デバイス・リセットは、I/Oモジュールに接続されたI/Oデバイスをリセットするために使用される。これらのリセットは装置に依存し、装置が接続されているI/Oモジュールによって与えられる。

他のクラスのリセットはソフト・リセットである。上で説明したように、ソフト・リセットは、システム10内の状態エンジンと一時的レジスタをクリアするが、これらはクロスリンク内のモード・ビットのような構成情報を変化させない。更に、ソフト・リセットは、またモジュール内のエ

ラー処理機構をクリアするが、これらはシステム・エラー・レジスタ898およびシステム故障アドレス・レジスタ865のようなエラー・レジスタを変化させない。

ソフト・リセットには目標が定まっているので、その結果、システムの必要な部分のみがリセットされる。例えば、モジュール相互接続部130がリセットされる必要があれば、CPU40はリセットされず、またI/Oモジュール110に接続されている装置もリセットされない。

ソフト・リセットには3つのユニークな特徴がある。1つは各ゾーンがそれ自身のリセットの発生に対して責任を負っていることである。1つのゾーン内の故障エラーまたはリセット・ロジックは、従って故障の発生していないゾーンでリセットを行うことを防止される。

第2の特徴は、ソフト・リセットが命令実行のシーケンスを乱さないことである。CPU40、40'、50、および50'はクロックとハード・リセットの組み合わせのみによってリセットさ

れる。更に、メモリ制御装置70、75、70'、および75'はハード・リセットに取り付けたCPU命令にサービスを行うのに必要なそれらの状態エンジンとレジスタを有している。従って、ソフト・リセットはソフトウェアの実行にとって透明である。

第3の特徴は、ソフト・リセットの範囲、すなわちソフト・リセットによって影響を与えられるシステム10内の構成要素の数がシステム10のモードと最初のリセットに対する要求によって決まるということである。完全デブプレックス・モードの場合、CPUモジュール30で開始されるソフト・リセットに対する要求によって、ソフト・リセットがCPUモジュールの全ての構成要素およびモジュール相互接続部130と132に取り付けられた全てのファイアウォール1000と1010に対して行われる。従って、モジュール相互接続部130と132によってサービスを受ける全てのモジュールはそれらの状態エンジンと一時的レジスタのリセットを有している。これに

よって、過渡的なエラーによって発生される全ての問題のシステム・パイプラインがクリアされる。システム10は、デブプレックス・モードにあるので、ゾーン11'はゾーン11の行っている全ての事柄を行う。従って、CPUモジュール30'は、CPUモジュール30と同時に、ソフト・リセットに対する要求を出す。ゾーン11'内のソフト・リセットは、ゾーン11内のソフト・リセットと同じ効果を有している。

しかし、システム10がマスタ/スレーブ・モードにありCPUモジュール30'がスレーブ・モードにある場合、CPUモジュール30で始まるソフト・リセットに対する要求は、予期できるように、CPUモジュール30の全ての構成要素とモジュール相互接続部130と132に取り付けられた全てのファイアウォール1000と1010に対してソフト・リセットを出す。更に、ソフト・リセットに対する要求は、クロスリンク90と90'、クロスリンク・ケーブル25およびクロスリンク90'と95'を介してCPUモジュール

30'に出される。一部のモジュール相互接続部130と132はソフト・リセットを受け取る。この同じ構成の場合、CPUモジュール30'から開始されるソフト・リセットに対する要求は、メモリ制御装置70'と75'およびクロスリンク90'と95'に一部のみリセットする。

ソフト・リセットは、「CPUソフト・リセット」と「システム・ソフト・リセット」を有する。CPUソフト・リセットは、要求を最初に出したCPUモジュールの状態エンジンに影響を及ぼすソフト・リセットである。システム・ソフト・リセットは、モジュール相互接続部とこれに直接取付けられた構成要素に対するソフト・リセットである。CPUモジュールは、常にCPUソフト・リセットを要求することができる。システム・ソフト・リセットは、CPUを要求するクロスリンクがデブプレックス・モード・マスター/スレーブ・モード、またはオフ・モードにある場合にのみ、要求することができる。スレーブ・モードにあるクロスリンクは、他方のゾーンからシステム

・ソフト・リセットを与えられ、それ自身のモジュール相互接続部に対してシステム・ソフト・リセットを発生する。

CPUソフト・リセットは、エラーの状態に続いていてCPUのパイプラインをクリアする。CPUパイプラインは、メモリ相互接続部80と82、メモリ制御装置75および75内のラッチ(図示せず)、DMAエンジン800およびクロスリンク90と95を有する。CPUソフト・リセットは、またDMAまたはI/Oのタイムアウトに続いて発生することもできる。DMAまたはI/Oのタイムアウトは、I/Oデハイスが特定の時間間隔内にDMAまたはI/Oの要求に対して応答しない場合に発生する。

第21図は、CPUモジュール30および300'からI/Oモジュール100、110、100'、および110'とメモリ・モジュール60および60'に対するリセット線を示す。CPUモジュール30は、何時電源が印加されたかを示すDCK信号を受け取る。リセットをイニシャライズ

するのはこの信号である。CPUモジュール30'は、その電源から同じ信号を受取る。

1つのシステム・ハード・リセット線は、各I/Oモジュールに送られ、1つのシステム・ソフト・リセットは3つのI/Oモジュールの全てに送られる。1つのハード・リセットが各モジュールに対して必要である理由は、システム・ハード・リセット線がシステム10から個々のI/Oモジュールを取除くのに使用されるからである。各システム・ソフト・リセットに対してI/Oモジュールを3つに制限しているのは、単にローディングを考慮しているからに過ぎない。更に、1つのクロック・リセット線が全てのI/Oモジュールとメモリ・モジュールに送られる。1つのモジュールについて1つの線を使用する理由は、負荷を制御することによってスキューを制限するためである。

第22図は、リセットに関連するCPUモジュール30の構成要素を示す。CPU40と50は、それぞれクロック発生装置2210と2211を

有している。メモリ制御装置70と75は、それぞれクロック発生装置2220と2221を有し、クロスリンク90と95は、それぞれクロック発生装置2260と2261を有する。クロック発生装置は、システム・クロック信号を個々のモジュールによって使用するために分割する。

メモリ制御装置70は、リセット制御回路2230とソフト・リセット要求レジスタ2235を有する。メモリ制御装置75は、リセット制御回路2231とソフト・リセット要求レジスタ2236を有する。

クロスリンク90は、ローカル・リセット発生装置2240とシステム・リセット発生装置2250の両方を有している。クロスリンク95は、ローカル・リセット発生装置2241とシステム・リセット発生装置2251を有している。クロスリンクの「ローカル」部分は、このクロスリンクがスレーブ・モードにある場合に、CPUモジュールと共に残っているこのクロスリンクの部分であり、従って、シリアル・レジスタ、および幾つか

のバラレル・レジスタを有している。クロスリンクの「システム」部分は、モジュール相互接続部130と132（または130'と132'）とクロスリンク・ケーブル25にアクセスするために必要であるクロスリンクのその部分である。

ローカル・リセット発生装置3340と2241は、それぞれクロスリンク90と95のローカル・リセット制御回路2245と2246にハードおよびソフト・リセット信号を送ると共に、それぞれメモリ制御装置70と75のリセット制御回路2230と2231にハードおよびソフト・リセット信号を送ることによって、CPUモジュール30に対してリセットを発生する。ローカル・クロスリンク・リセット制御回路2245と2246は、それらの状態エンジン、転送すべきデータを記憶しているラッチおよびそれらのエラー・レジスタをリセットすることによって、ソフト・リセット信号にตอบสนองする。これらの回路は、ソフト・リセットに対して行うのと同じ動作を行い、またエラー・レジスタと構成レジスタをリセットするこ

とによって、ハード・リセット信号にตอบสนองする。リセット制御回路2230と2231は、同じ方法でハードおよびソフト・リセット信号にตอบสนองする。

更に、ローカル・リセット発生装置2240は、モジュール相互接続部130と132を介して、I/Oモジュール100、110および120にクロック・リセット信号を送る。I/Oモジュール100、110および120は、以下で述べる方法でそれらのクロックをリセットするため、クロック・リセット信号を使用する。ソフト・リセット要求レジスタ2235と2236は、それぞれローカル・リセット発生装置2240と2241にソフト要求信号を送る。

クロスリンク90と95のシステム・リセット発生装置2245と2251は、それぞれモジュール相互接続部130と132を介してI/Oモジュール100、110、および120にそれぞれシステム・ハード・リセット信号とシステム・ソフト・リセット信号に送る。I/Oモジュール

100、110、および120は、CPUデータまたは命令に依存する全てのレジスタをリセットすることによってソフト・リセット信号にตอบสนองする。これらのモジュールは、ソフト・リセットが行なうのと同じレジスタをリセットし、また全ての構成レジスタをリセットすることによって、ハード・リセット信号にตอบสนองする。

更に、システム・リセット発生装置2250と2251は、またシステム・ソフトおよびシステム・ハード・リセット信号を各クロスリンクのシステム・リセット制御回路2255と2256に送る。システム・リセット制御回路2255と2256は、ローカル・ソフトおよびローカル・ハード・リセット信号に対するローカル・リセット制御回路の応答と同じ方法でシステム・ソフト・リセット信号とシステム・ハード・リセット信号にตอบสนองする。

メモリ制御装置70と75は、CPU40と50がそれぞれ適当なコードをソフト・リセット要求レジスタ2235と2236にそれぞれ書込

み場合に、クロスリンク90と95にそれぞれソフト・リセットを発生させる。ソフト・リセット要求レジスタ2235と2236は、ソフト・リセット要求信号をローカル・リセット発生装置2240と2241に送る。符号化エラー信号は、メモリ制御装置70からローカル・リセット発生装置2240と2241に送られる。

システム・ソフト・リセットは、データと制御信号が送られるのと同じデータ経路に沿ってゾーン間に送られる。従って、データとアドレスに対するのと同じ遅延を等しくする原理が使用され、リセットはほぼ同時に2つのゾーンの全ての構成要素に到達する。

ハード・リセットは、適当なコードをローカル・ハード・リセット・レジスタ2243に書き込むCPU40と50またはDC OK信号によって発生されるパワーアップ・リセットに対する要求によって発生される。

クロスリンク90の同期回路2270は、DC OK信号が同時にローカルおよびリセット発生装

置2240、2250、2241および2251の全てに行き渡ることを保証するため、適当な遅延要素を有している。

事実、リセットの同期は、システム10では非常に重要である。これは、リセット信号がクロスリンクで始まるからである。このようにして、リセットはほぼ同期して異なったモジュールとこれらのモジュール内の異なった要素に到達するように送られることができる。

第21図と第22図の構造を理解することによって、異なったハード・リセットの実行をよりよく理解することができる。パワーアップ・リセットはシステム・ハード・リセットとローカル・ハード・リセットおよびクロック・リセットの両方を発生する。一般的に、クロスリンク90、95、90'および95'は最初はクロスリンク・オフモードと再同期オフ・モードの両方の状態にあり、両方のゾーンはクロック・マスターシップを表明する。

CPU/MEM故障リセットは、メモリ制御装

置70、75、70'および75'がCPM/MEMの故障を検出する時は何時でも自動的に動作される。符号化エラー・ロジックはエラー・ロジック2237と2238から両方のクロスリンク90と95に送られる。故障が発生したCPUモジュールは、そのクロスリンクをスレーブ状態にセットし、他方のCPUモジュールのクロスリンクをマスター状態にセットすることによって、システム10から取り除かれる。しかし、故障が発生していないCPUモジュールは、リセットを経験しない。その代わり、これはシリアル・クロスリンク・エラー・レジスタ(図示せず)内のコードを介して、他方のモジュールの故障を知らされる。CPU/MEM故障リセットは、故障したCPUモジュールを有するゾーンに対するクロック信号とそのモジュールに対するローカル・ソフト・リセットによって構成される。

再同期リセットは、基本的にはローカル・ハード・リセットとクロック・リセットを有するシステム・ソフト・リセットである。この再同期リセ

ットは、2つのゾーンをロックステップ同期の状態にするために使用される。ゾーン11と11'が同期されていなかった一定の期間の後、もしCPUレジスタの記憶された状態を含むメモリ・モジュール60と60'の内容が相互に等しくセットされれば、これらのゾーンがデュプレックス・モードを再び開始することができるように、再同期リセットが使用されてこれらのゾーンを互換性のある構成にする。

再同期リセットは、基本的にはCPUハード・リセットとクロック・リセットである。再同期リセットは、再同期・リセット・アドレスを並列クロスリンク・レジスタの1つに書き込むソフトウェアによって動作される。この時、一方のゾーンは、クロスリンク・マスター/再同期マスター・モードでなければならない。他方のゾーンは、クロスリンク・スレーブ/再同期スレーブ・モードでなければならない。そこでリセットが両方のゾーンで同時に行われ、これは、とりわけ4つのクロスリンク全てをデュプレックス・モードにセットする。

再同期リセットは、システム・ソフト・リセットではないため、1/Oモジュールはリセットを受取らない。

システム10の好適な実施例は、またクロック・リセット信号がコンフォーミング (conforming) クロックをリセットせず、非コンフォーミング・クロックのみをリセットすることを保証する。この理由は、クロックがリセットされる場合はいつでも、これはクロックのタイミングを変更し、このタイミングはこんどはこのようなクロックでモジュールの動作に影響を及ぼすからである。もしモジュールが正しく実行され、このクロックが正しい位相であれば、その動作を変更することには必要であるばかりでなく無駄なことである。

第23図は、ノンコンフォーミング・クロックのみがリセットされることを保証する回路の好適な実施例である。第23図に示す回路は、第22図に示す対応するモジュールのクロック発生装置2210、2211、2220、2221、2260、および2261内に位置することが望

ましい。

好適な実施例の場合、異なるクロック発生装置2210、2211、2220、2221、2260、および2261は立上がり区間検出器2300、と位相発生装置2310を有している。立上がり区間検出器2300は、クロスリンク90と95からクロック・リセット信号を受取り、クロック・リセット信号の立上がり区間と同時に既知の持続期間を有するパルスが発生する。このパルスは、特定のモジュールに対する内部クロック信号と同様に位相発生装置2310に対する入力である。そのモジュールに対する内部クロック信号は、発振器システム200と200'から分配されたシステム・クロック信号から取出されたクロック信号である。位相発生装置2310は、クロック信号に対する異なる位相を形成する下方分割回路であることが望ましい。再循環シフト・レジスタのような位相発生装置2310に対する別の設計をまた使用することもできる。

立上がり区間検出器2300からの立上がり区

間パルスによって、位相発生装置2310は予め選択された位相を出力することが望ましい。従って、例えばもし位相発生装置2310が幾つかのステージを有する下方分割回路であれば、クロック・リセットの立上がり区間パルスは、そのステージに対して設定された入力であり、このステージは全ての他のステージに対して予め選択された位相とりセット入力を発生する。もし位相発生装置2310が既にこの位相を発生していれば、同期化クロック・リセット信号の存在は基本的に透明である。

このようにして組織されたりセットは、システム10の通常の実行に対して混乱を最小限に止めるように設計され、ドラスチックなアクションが必要とされる場合には、このドラスチックなアクションは命令実行の通常のシーケンスに割込みをかけることに止まる。このことは、従来のリセットが引起こす再同期化の問題のためにデュアルまたは多重ゾーン的环境下では特に重要である。従って、システム10で行っているようにハード・リ

セットの数を最小にすることが望ましい。

4. 図面の簡単な説明

第1図は、本発明を実施する故障許容コンピュータ・システムの好適な実施例のブロック図である。

第2図は、第1図の故障許容コンピュータ・システムを有する物理的ハードウェアを示す。

第3図は、第1図の故障許容コンピュータ・システムに示すCPUモジュールのブロック図である。

第4図は、第1図に示すコンピュータ・システムの相互に接続されたCPUモジュールと1/Oモジュールのブロック図を示す。

第5図は、第1図に示す故障許容コンピュータ・システムのメモリ・モジュールのブロック図を示す。

第6図は、第5図に示すメモリ・モジュールの制御ロジックの構成要素の詳細図である。

第7図は、第3図に示すCPUモジュールのプライマリ・メモリ制御装置の部分ブロック図を示

す。

第8図は、第3図のCPUモジュールのプライマリ・メモリ制御装置のDMAエンジンのらブロック図である。

第9図は、第3図のCPUモジュールのプライマリ・メモリ制御装置のエラー処理回路図である。

第10図は、第3図に示すCPUモジュールのクロスリンクの幾つかのレジスタの図である。

第11図は、第3図に示すCPUモジュールのクロスリンクに制御信号を流す構成要素のブロック図である。

第12図は、第3図に示すCPUモジュールのプライマリ・クロスリンクにデータとアドレス信号を流す構成要素のブロック図である。

第13図は、第3図に示すCPUモジュールのクロスリンクの状態を示す状態図である。

第14図は、第1図の故障許容コンピュータ・システムのタイミング・システムのブロック図である。

第15図は、第14図のタイミング・システム

によって発生されるクロック信号のタイミング図である。

第16図は、第14図に示すタイミング・システムの位相検出器の詳細図である。

第17図は、第1図のコンピュータ・システムのI/Oモジュールのブロック図である。

第18図は、第17図に示すI/Oモジュールのファイアウォールの構成要素のブロック図である。

第19図は、第1図のコンピュータ・システムのクロスリンク経路の構成要素の詳細図である。

第20A図ないし第20E図は第1図のコンピュータ・システムのデータ・フロー図である。

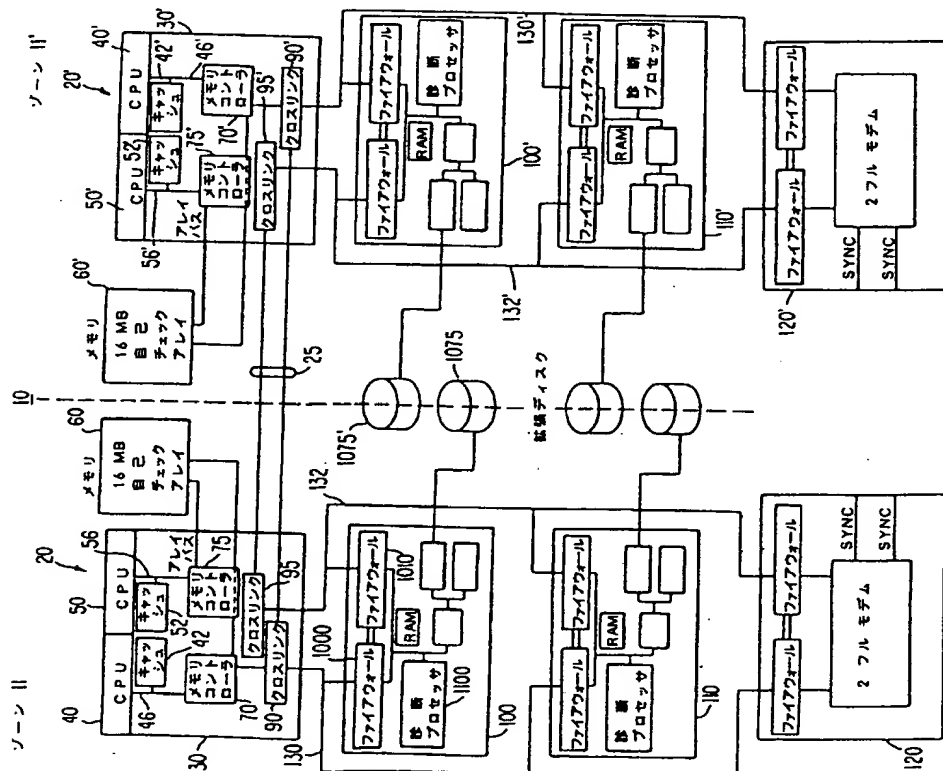
第21図は、リセット信号の流れを示すゾーン20のブロック図である。

第22図は、第3図に示すCPUモジュールのリセットに含まれる構成要素のブロック図である。

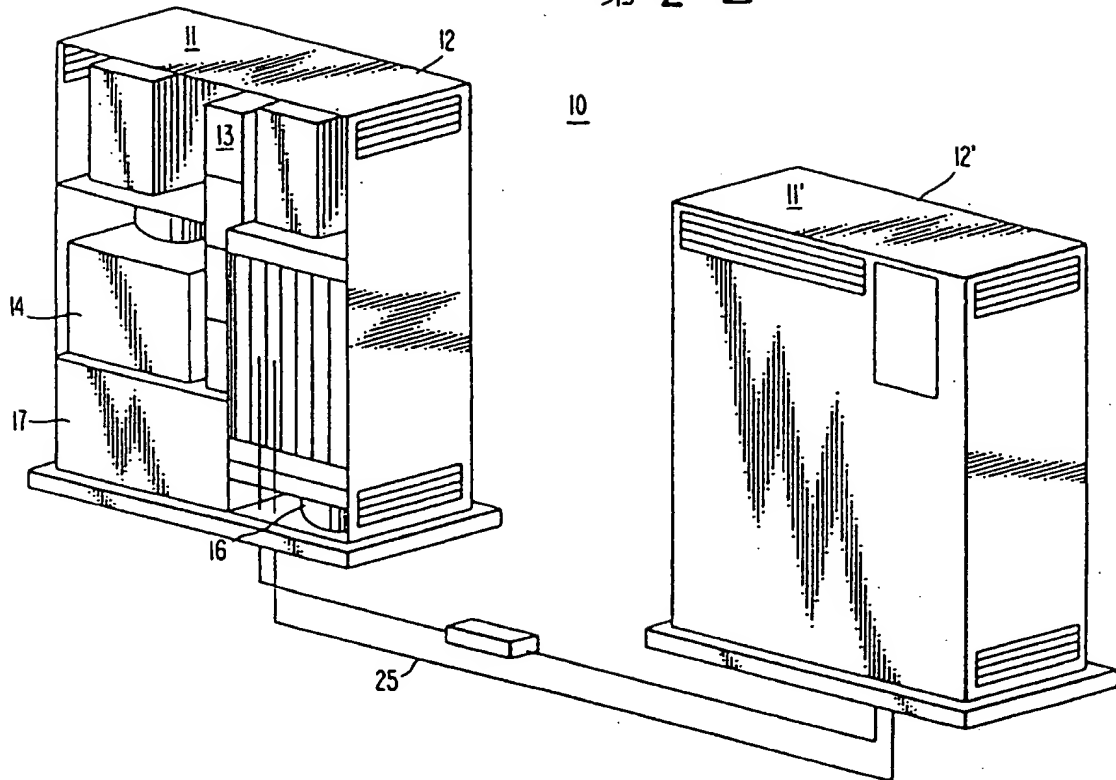
第23図は、クロック・リセット回路の図である。

図面の浄化(内容に変更なし)

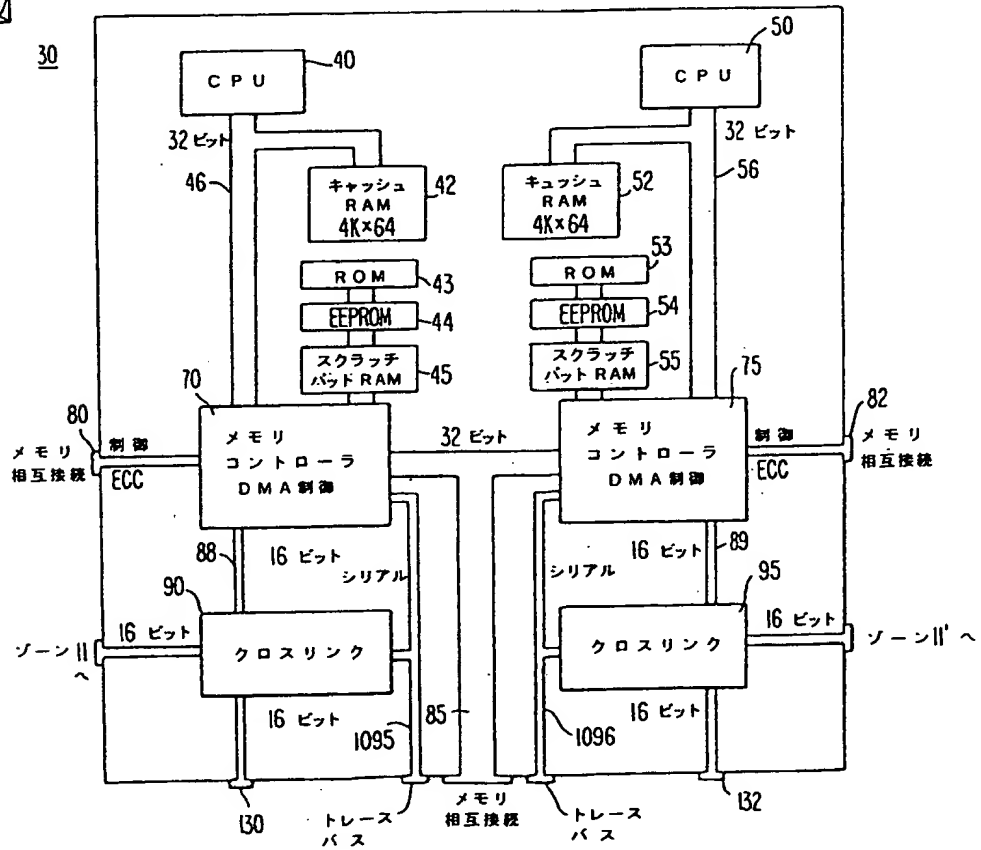
第 / 図



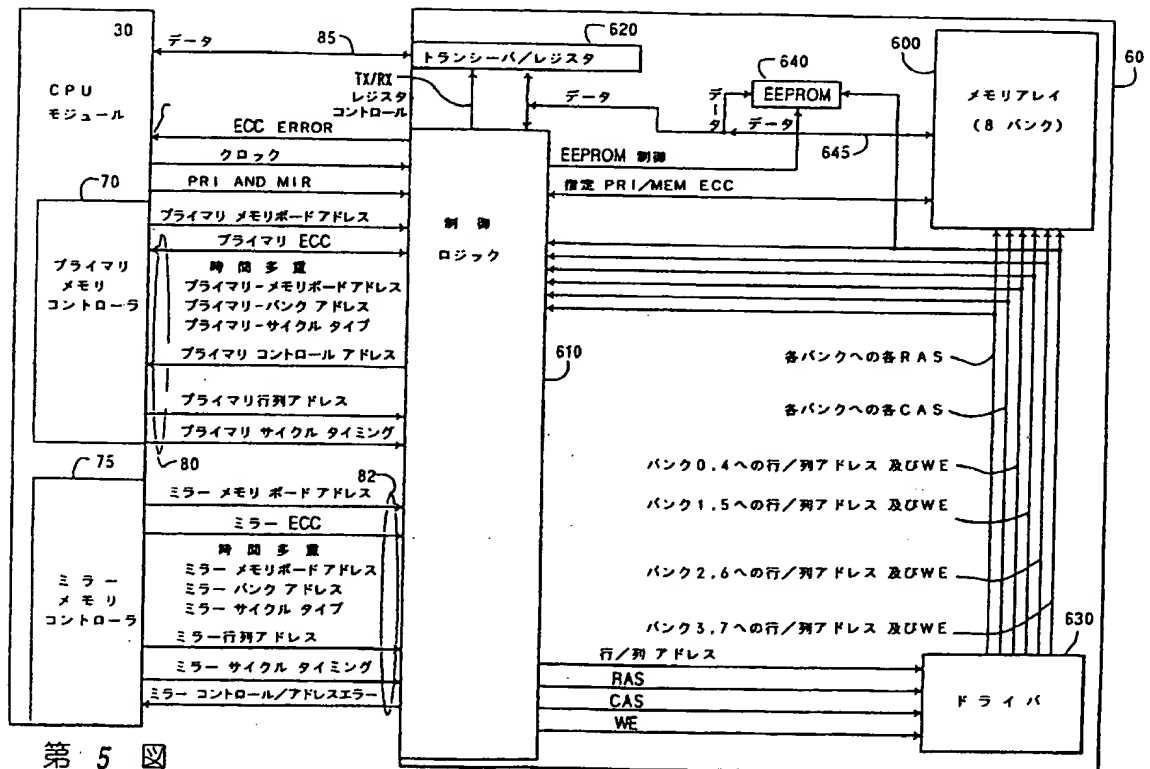
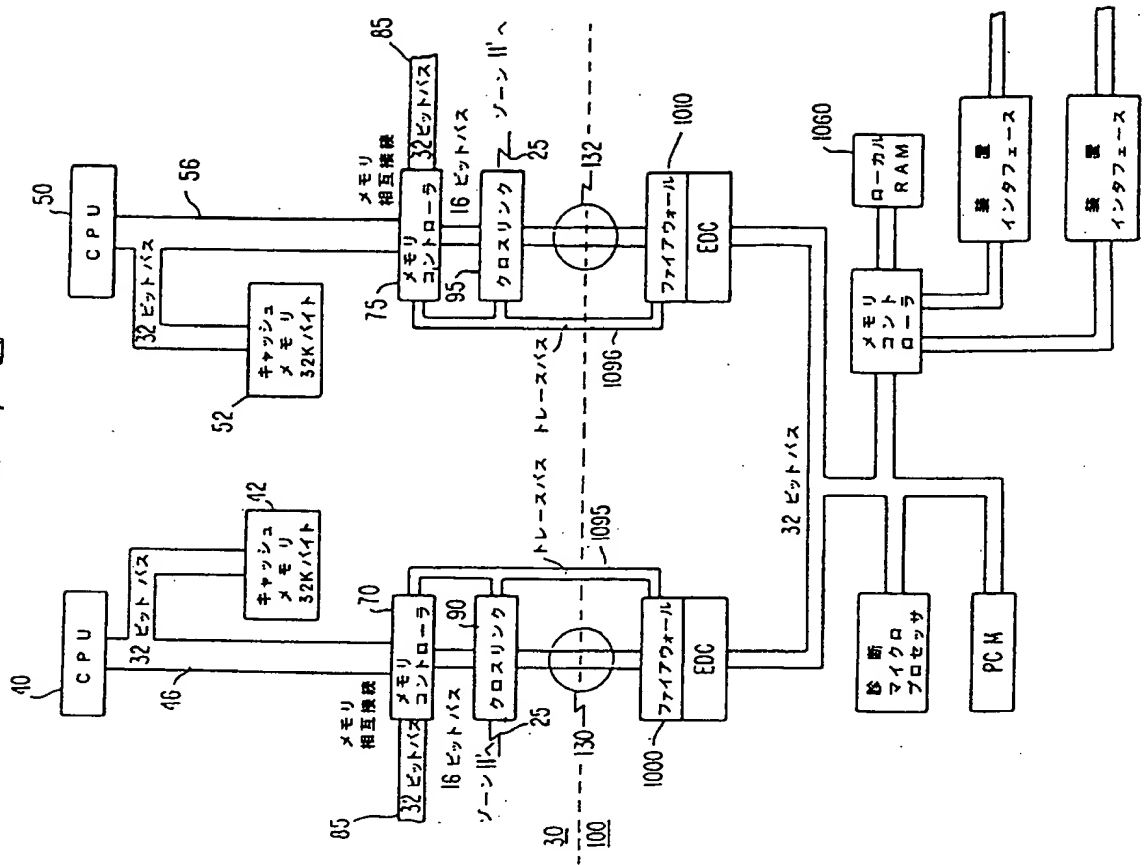
第 2 図



第 3 図

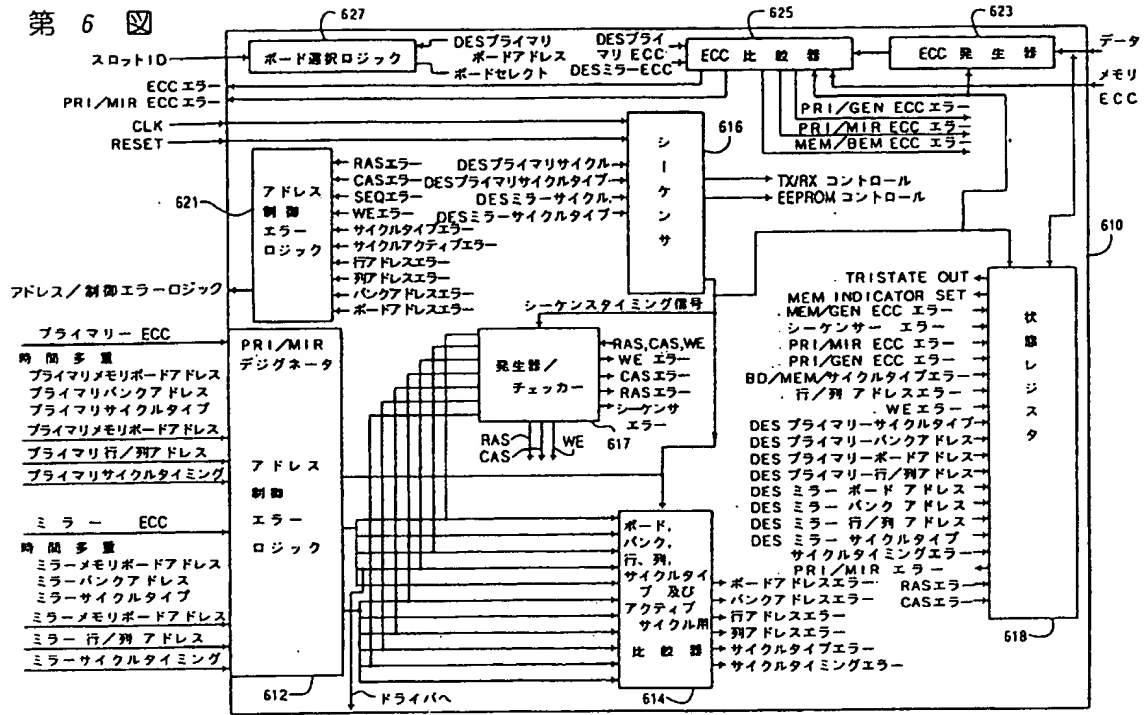


第 4 図

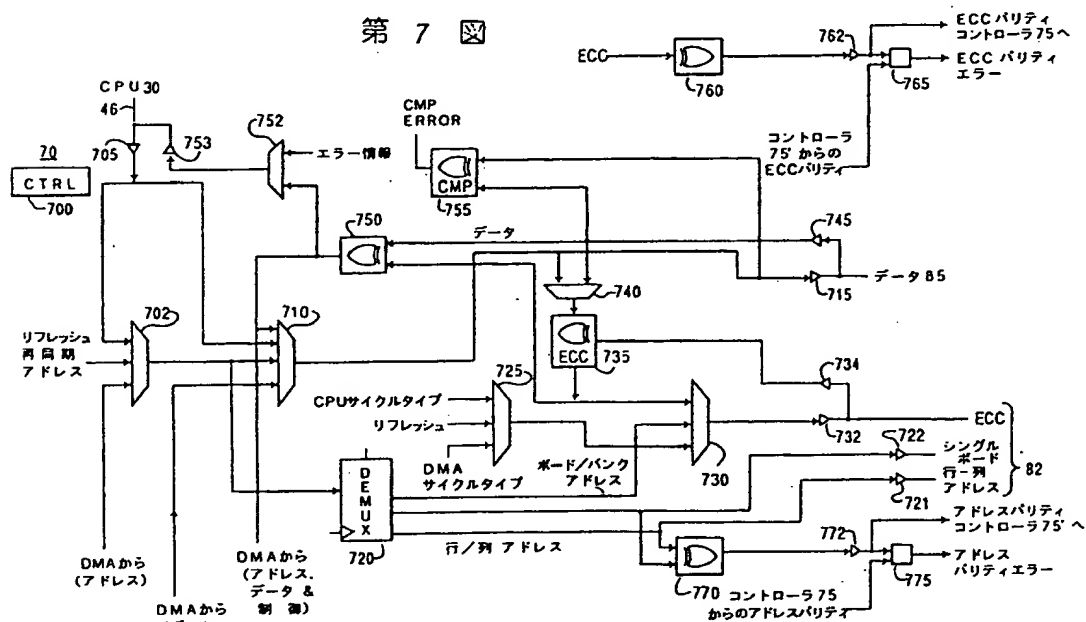


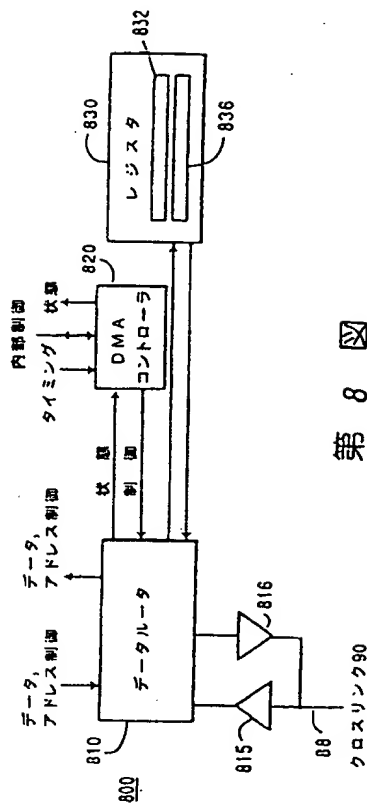
第 5 図

第 6 図

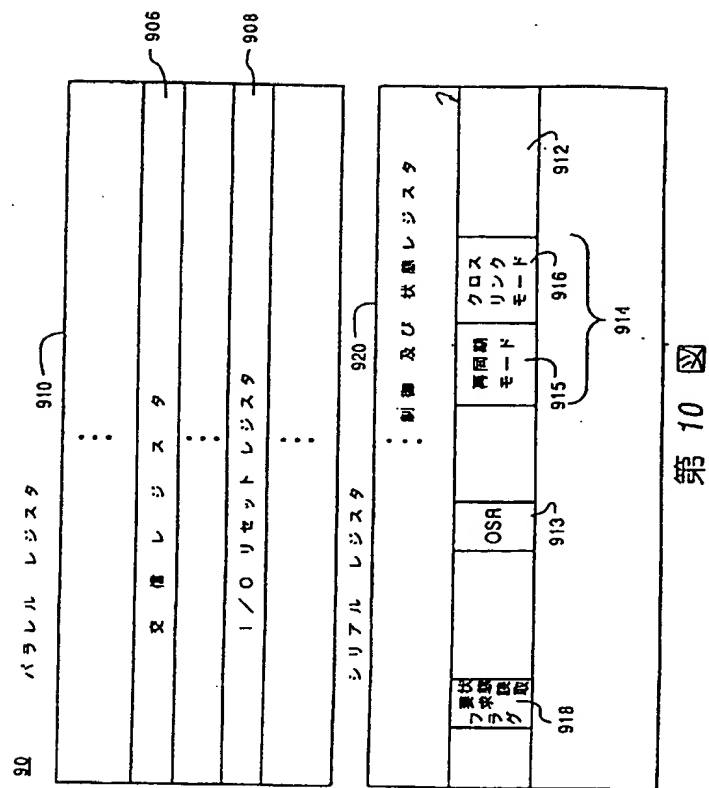


第 7 図

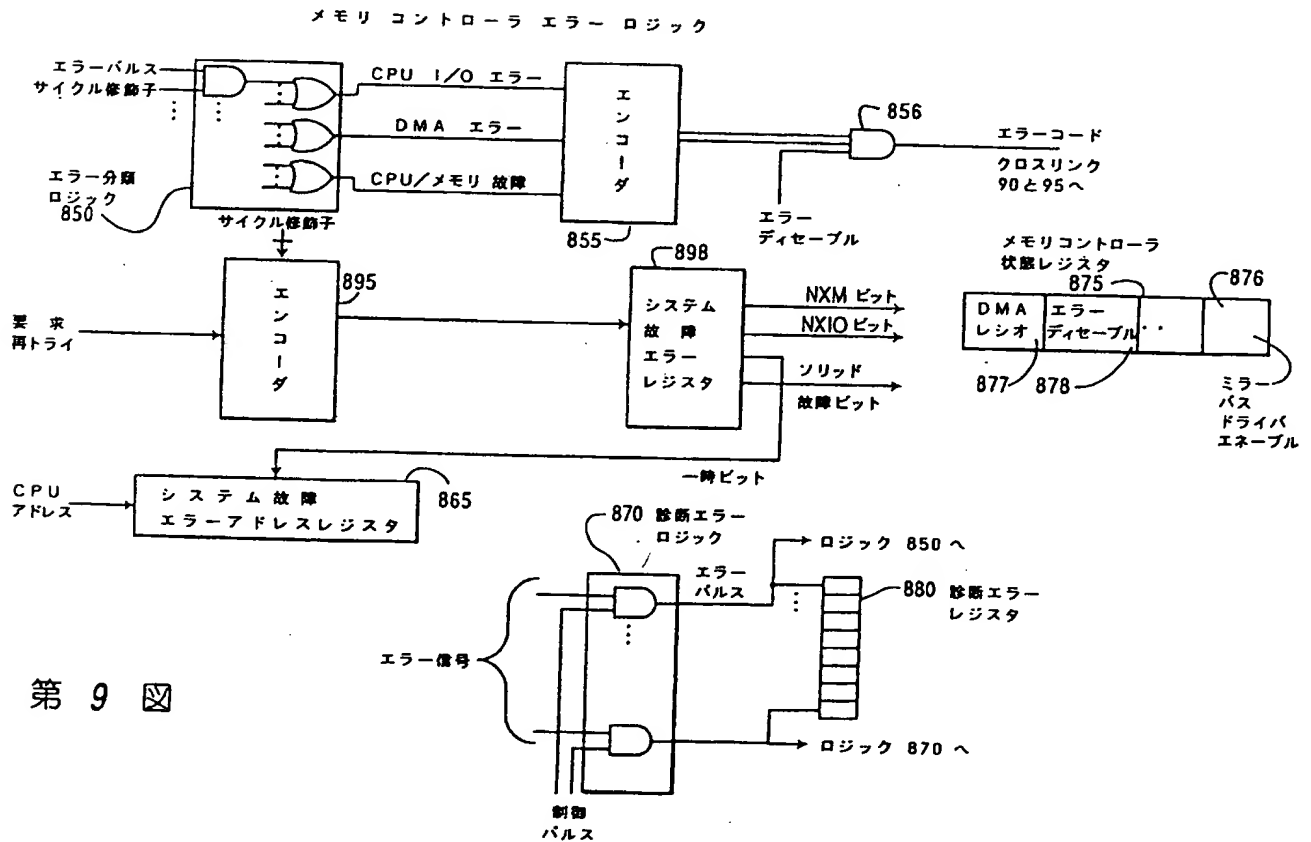




振

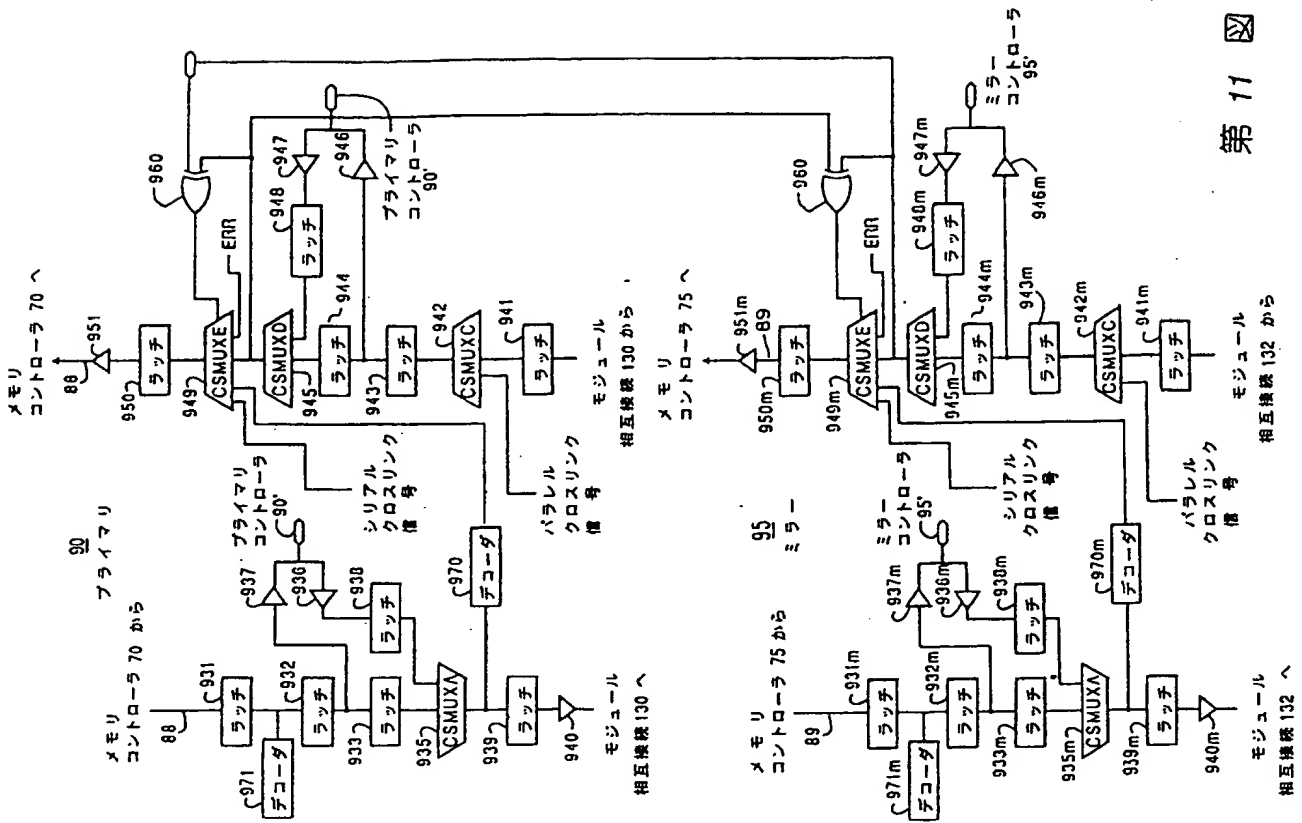


第 10 圖

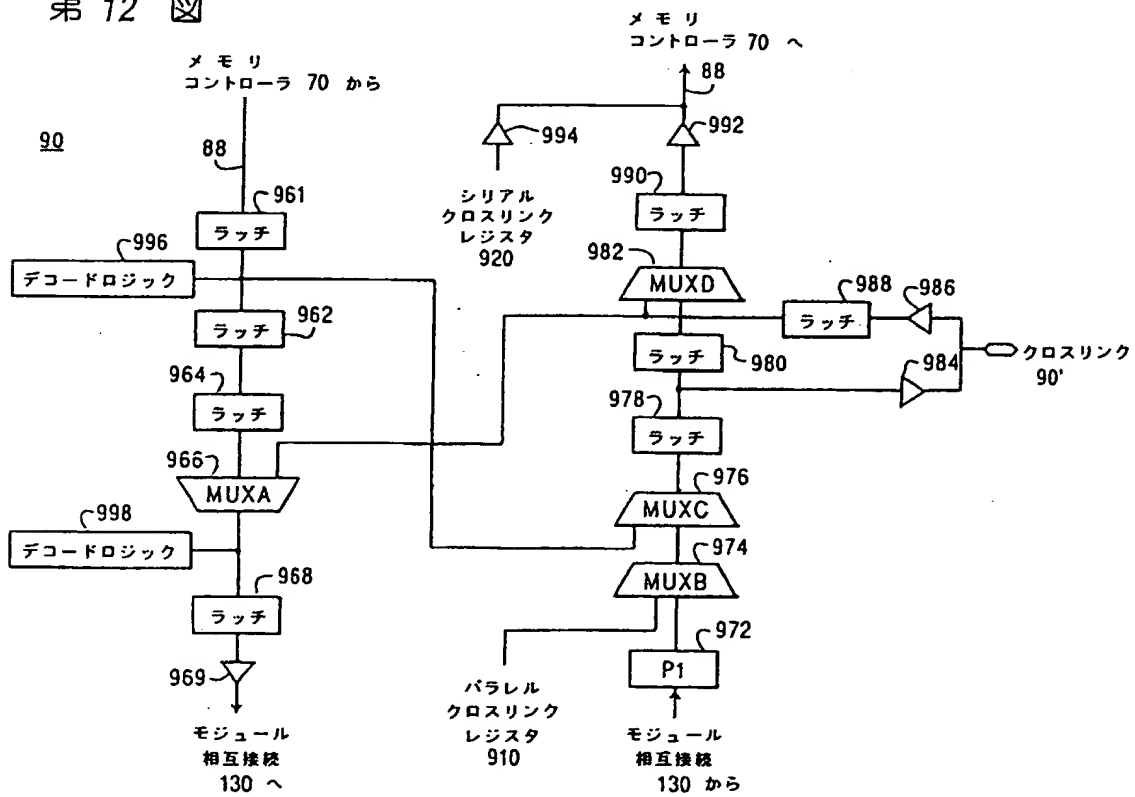


第 9 圖

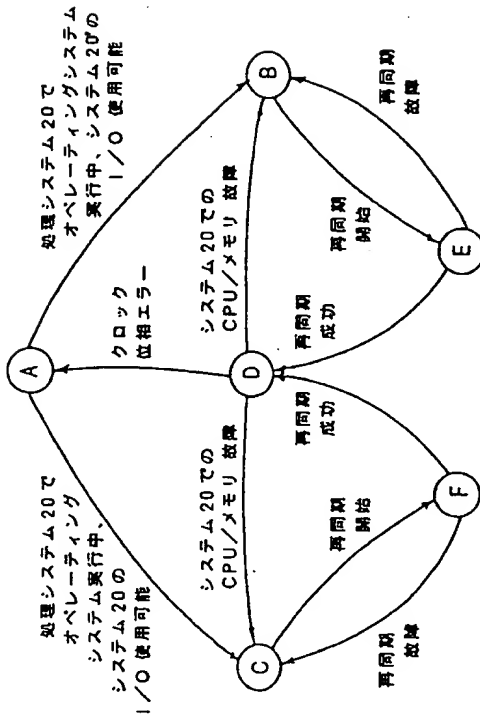
第 11



第 12



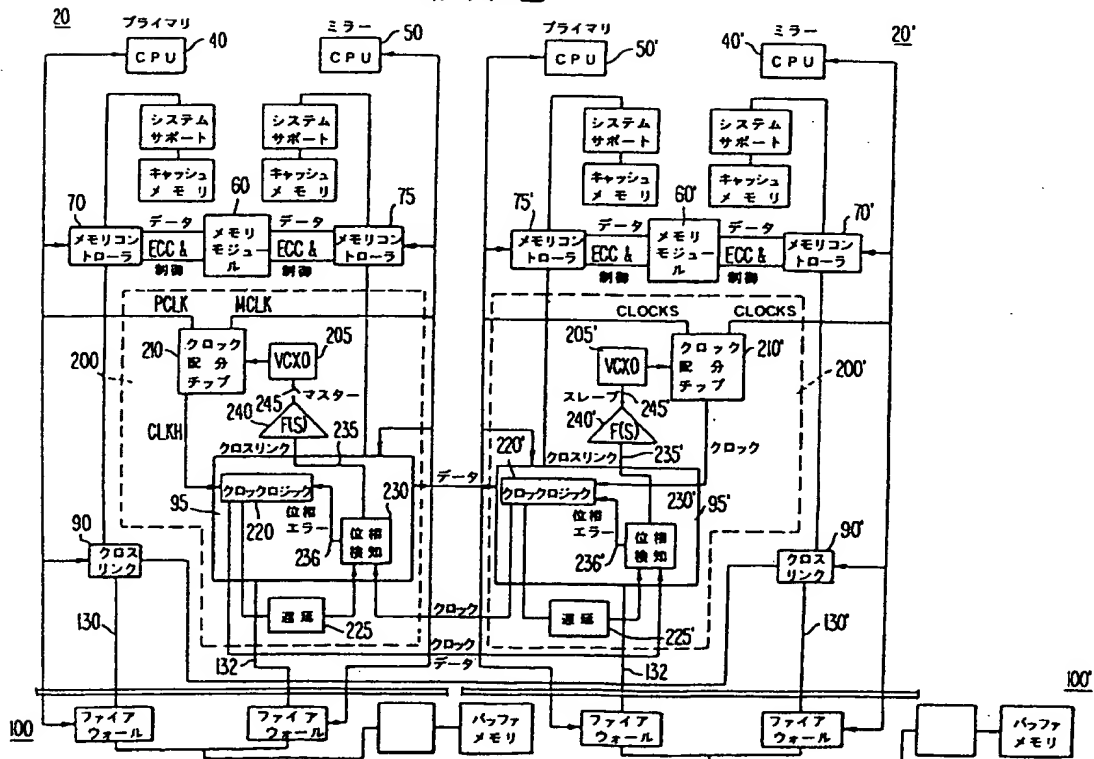
第 13 図



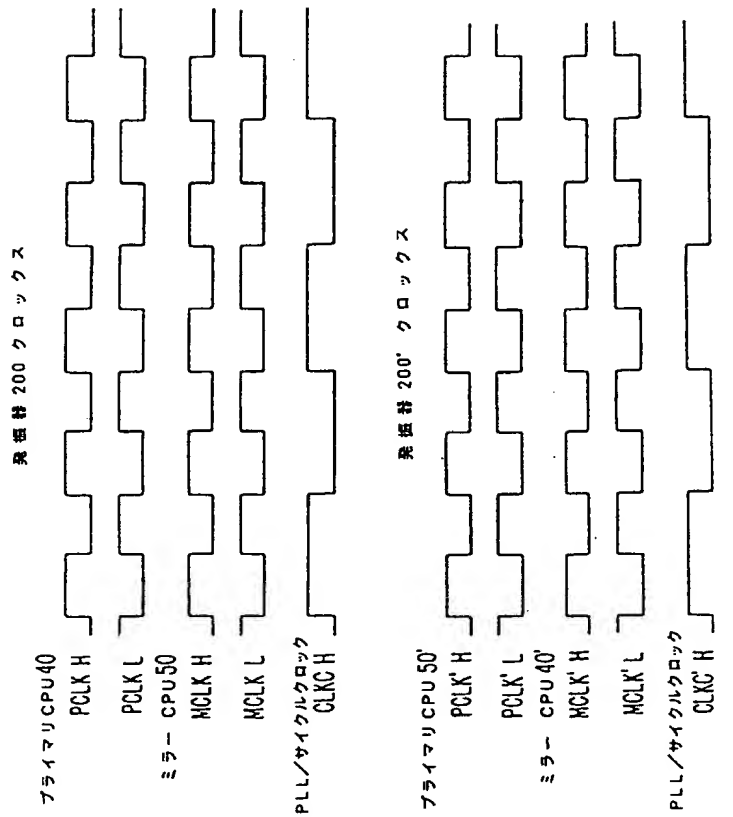
状態 処理 処理 システム システム 20 20'

状態	処理	処理	システム	システム
A	オフ	オフ	20	20'
B	マスター	スレーブ	20	20'
C	スレーブ	マスター	20	20'
D	全二重	全二重	20	20'
E	マスター	スレーブ	20	20'
F	スレーブ	マスター	20	20'

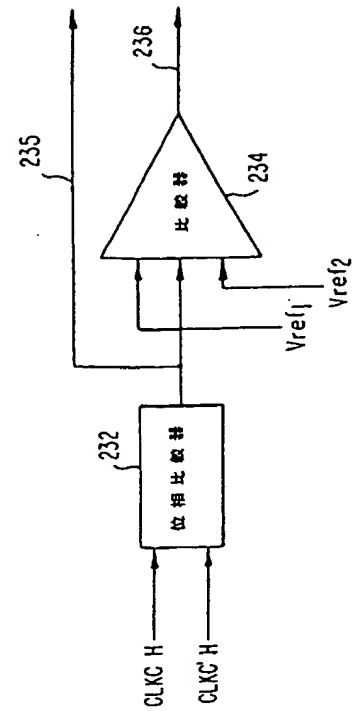
第 14 図



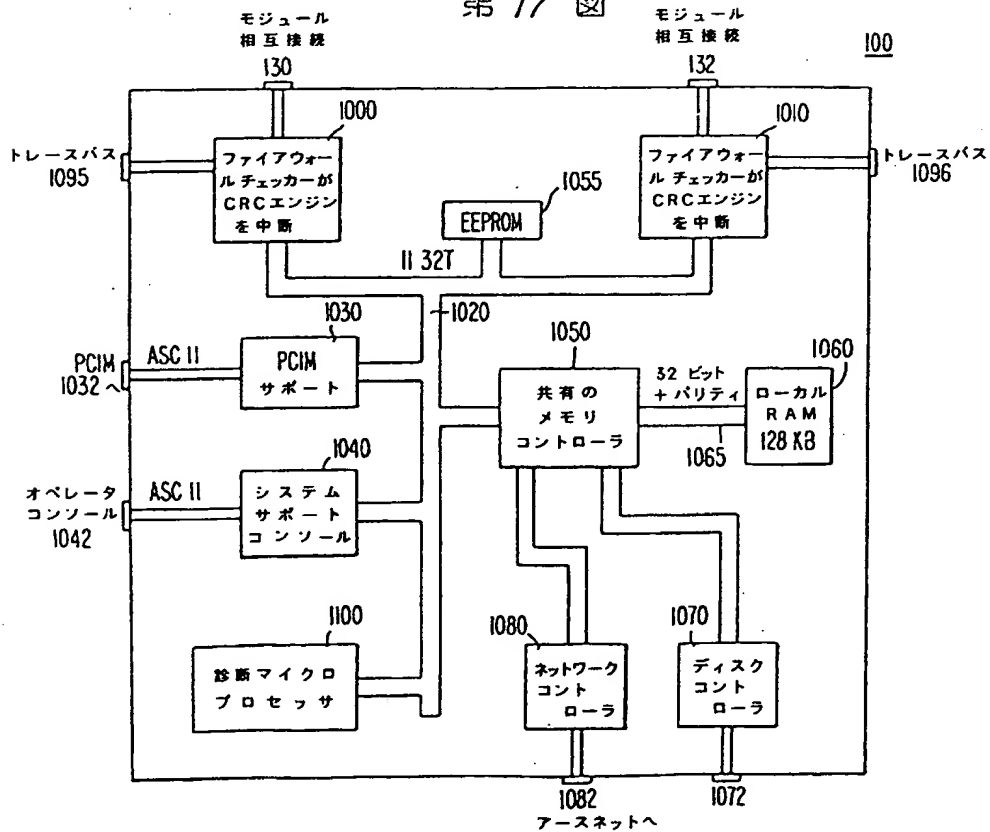
第 15



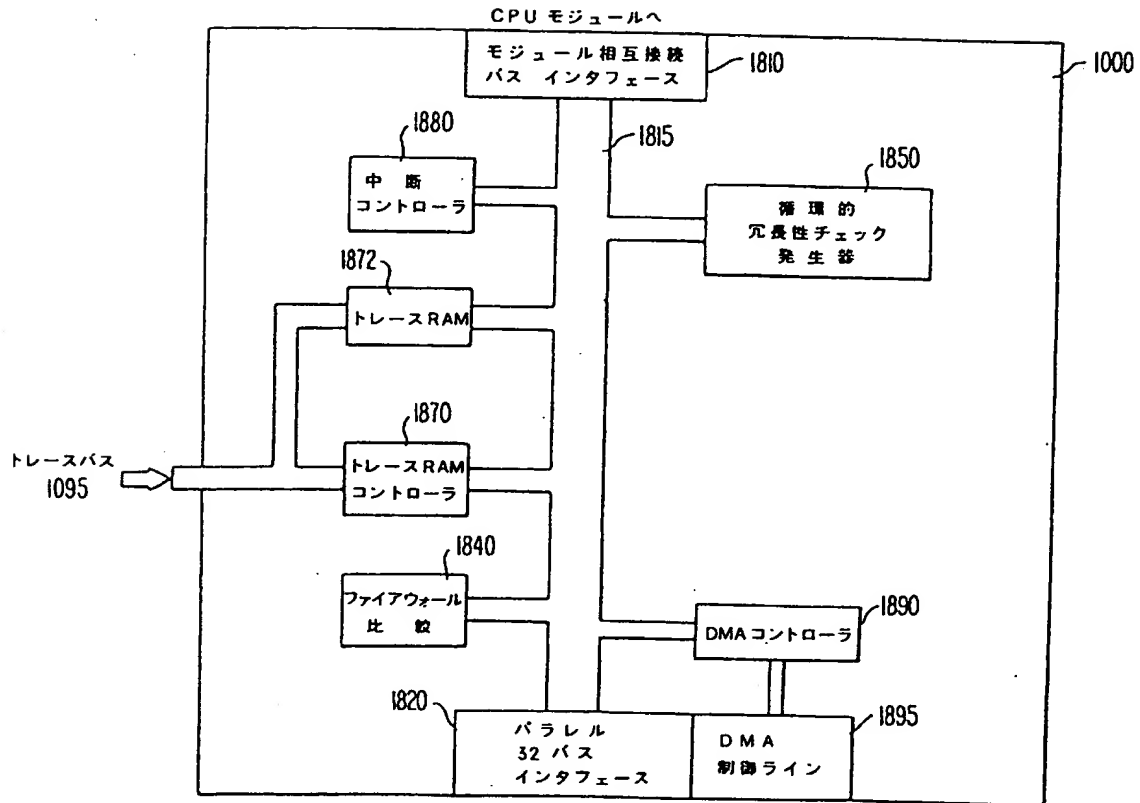
第 16



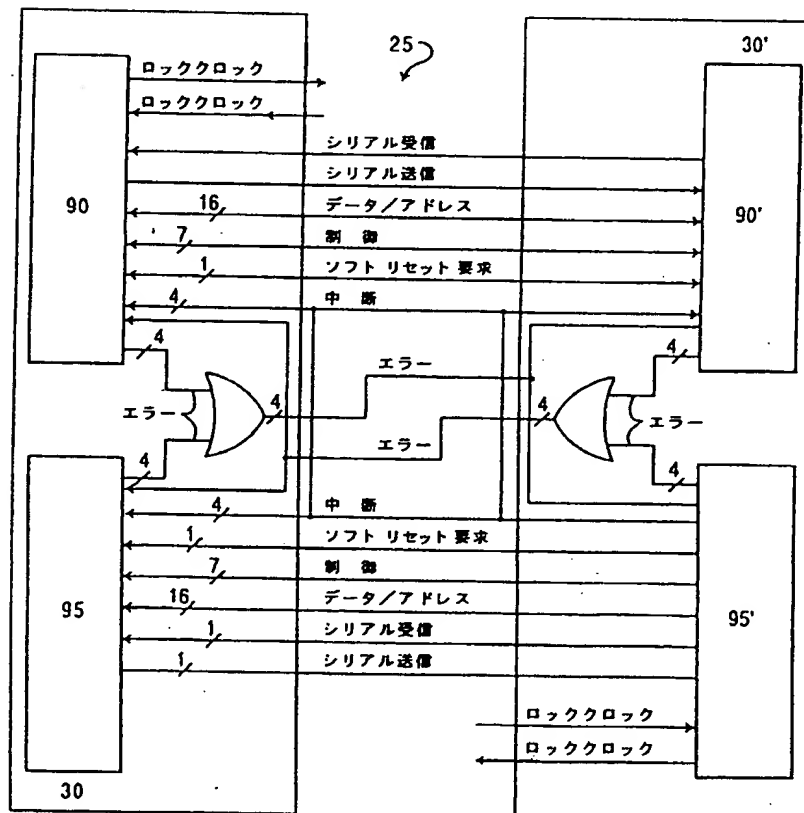
第 17



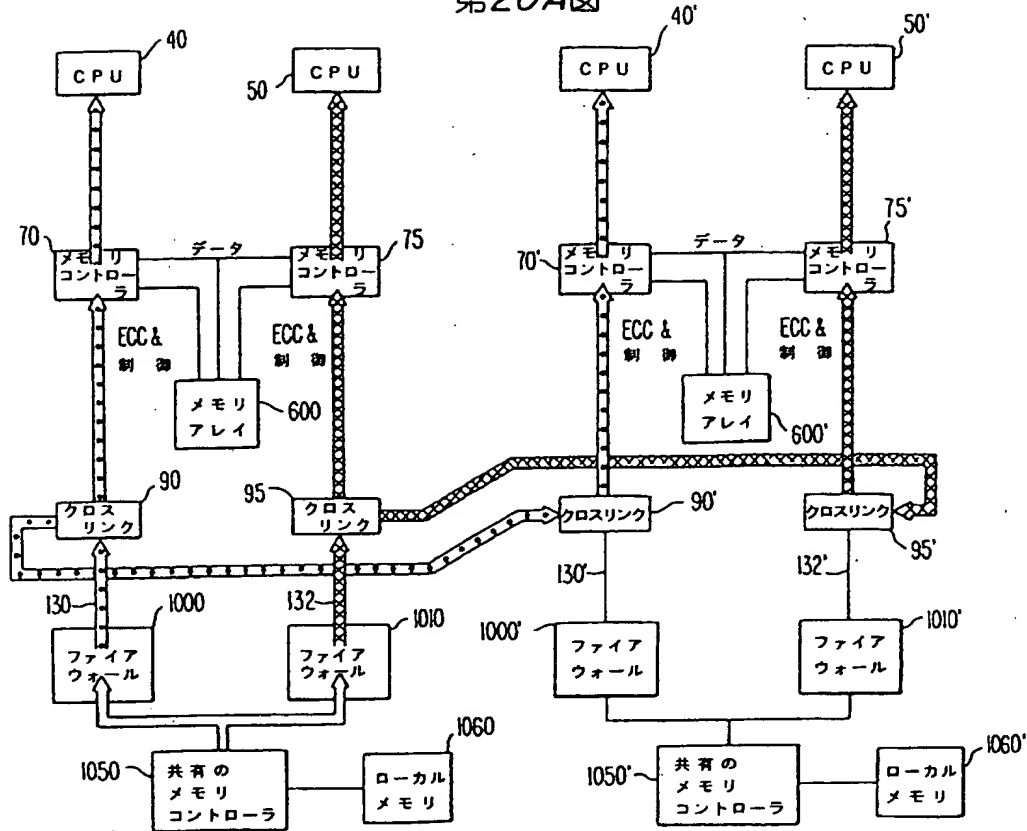
第 18 図



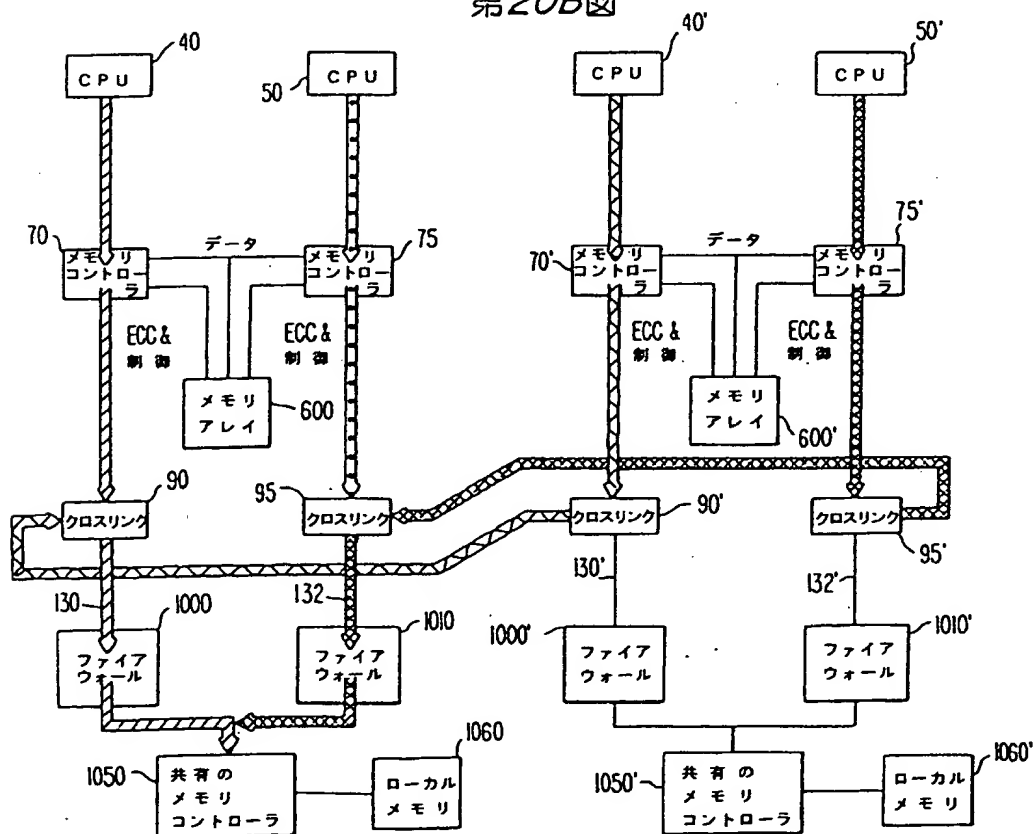
第 19 図



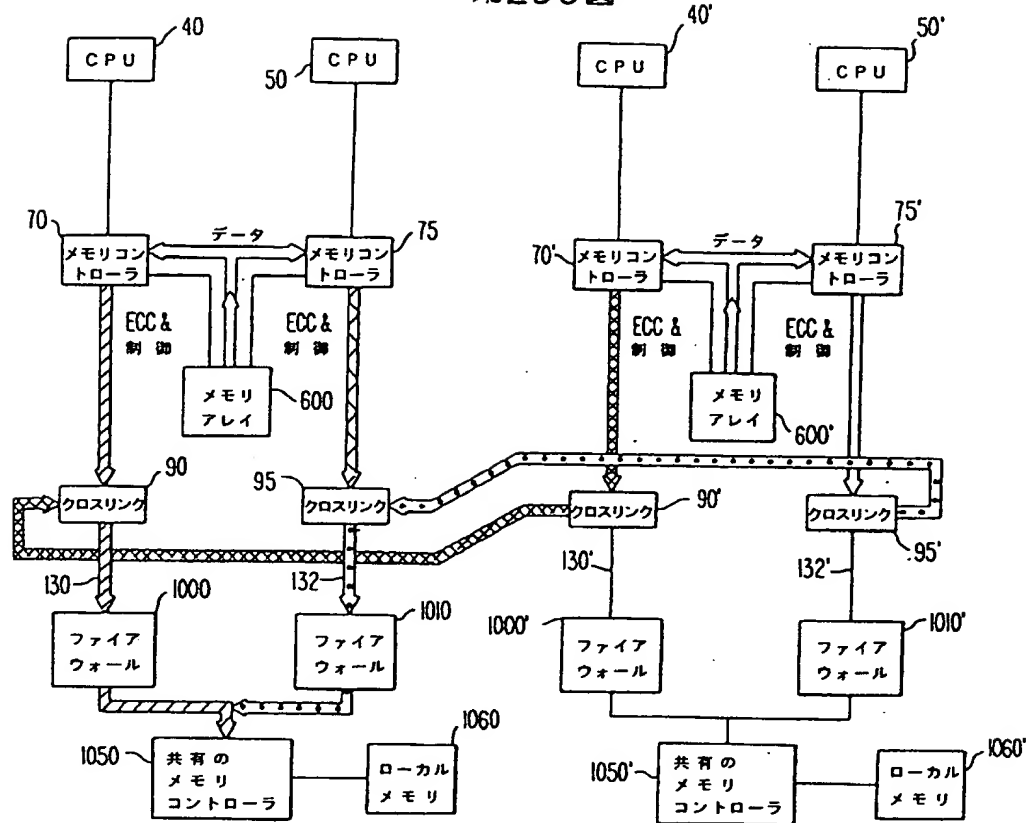
第20A図



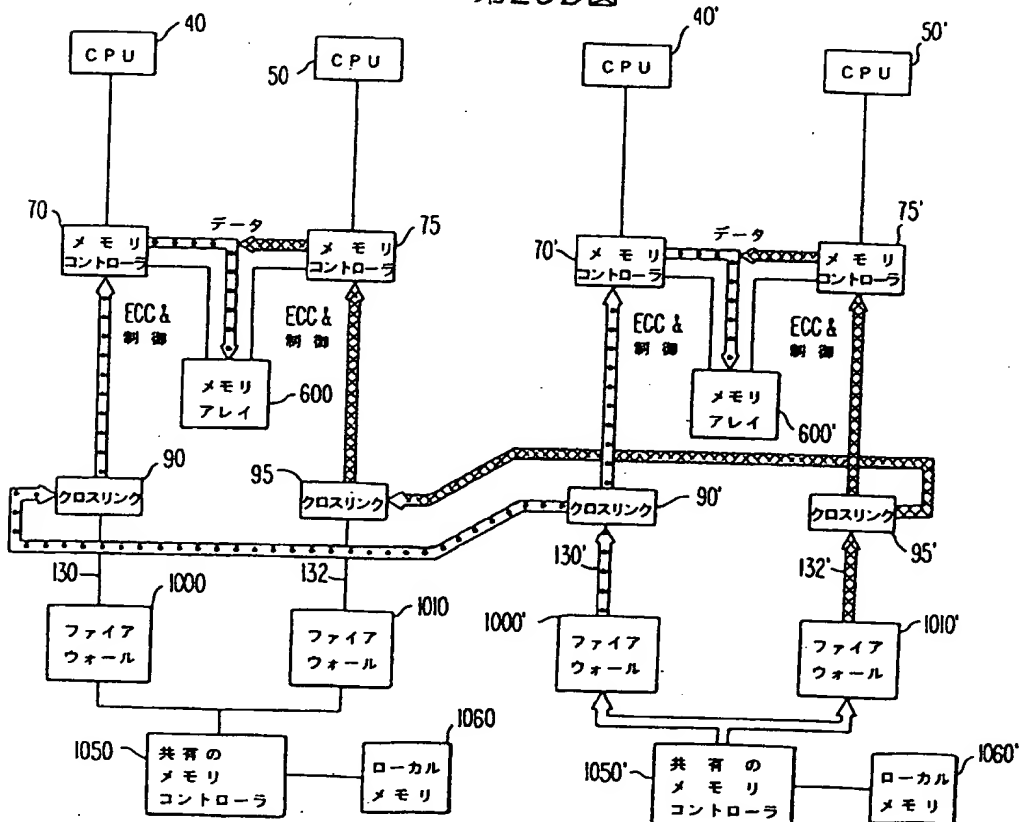
第20B図



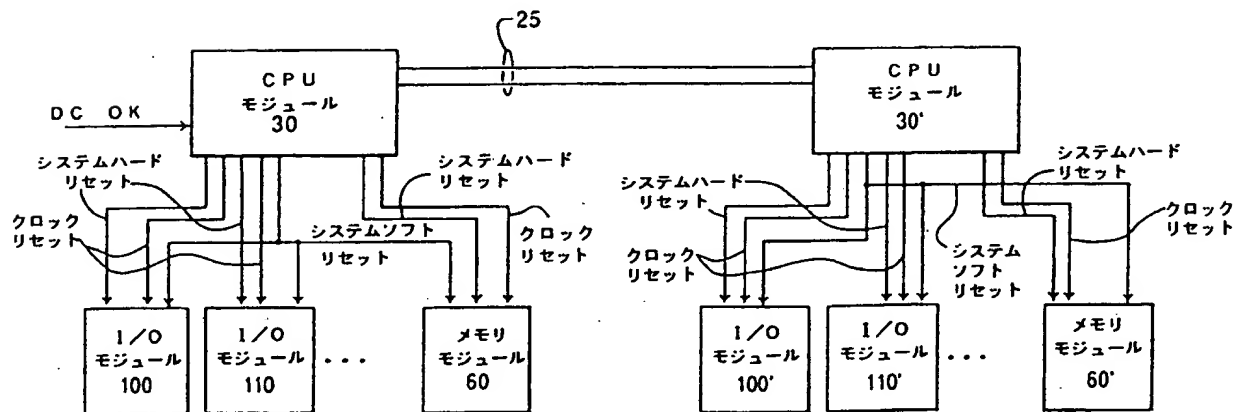
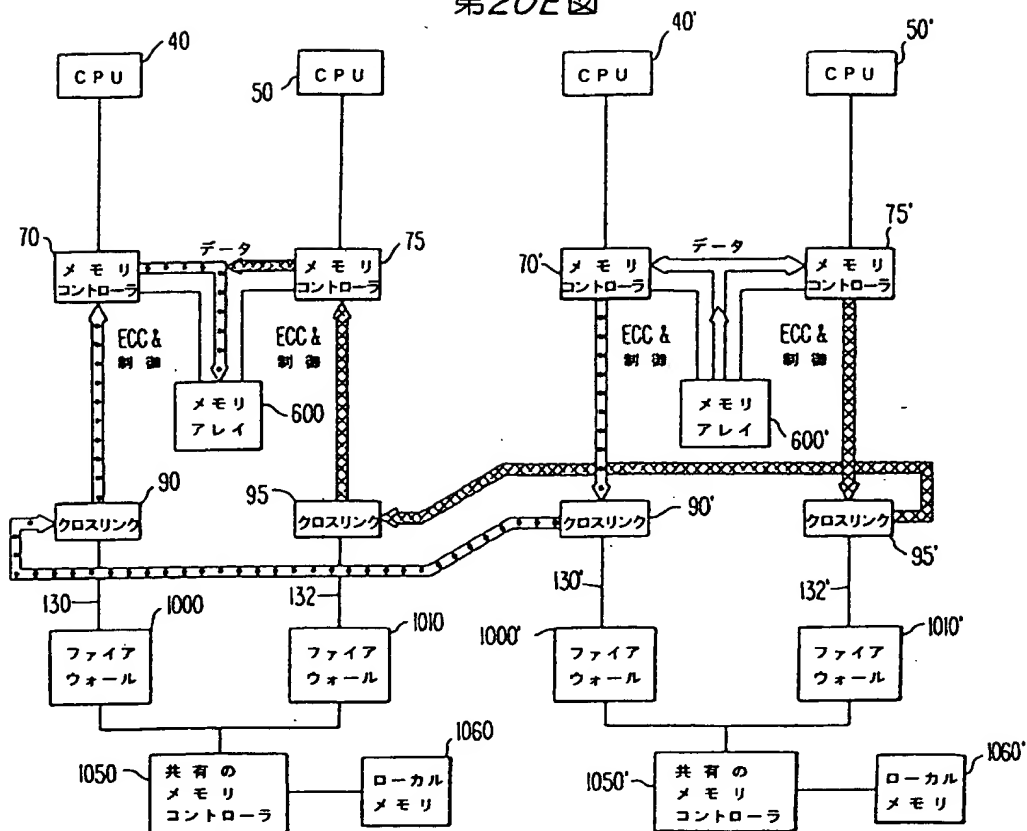
第20C図



第20D図

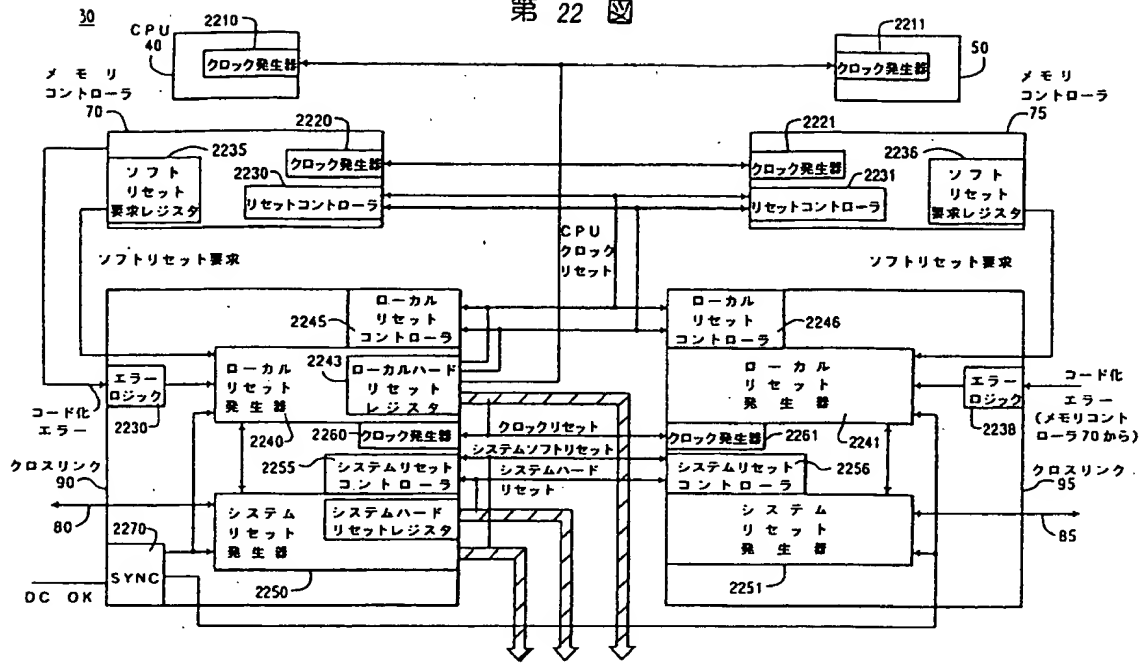


第20E図

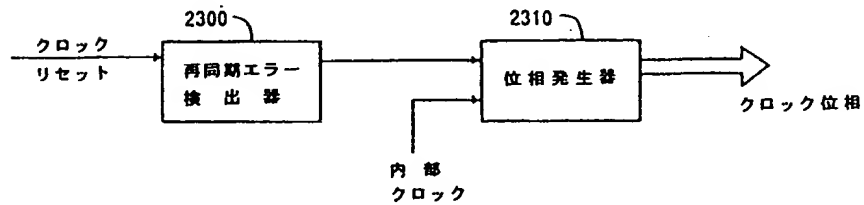


第 21 図

第 22 図



第 23 図



第1頁の続き

⑤Int. Cl.³

G 06 F 15/16

識別記号

4 2 0 S

庁内整理番号

6945-5B

- ⑦発明者 トーマス デイー ビ アメリカ合衆国 ニューハンプシャー州 03038 デリー
セツト オルセン ロード 21
- ⑦発明者 ジョン マンザー アメリカ合衆国 マサチューセッツ州 02146 ブルック
ライン ケント ストリート 131
- ⑦発明者 ミツチエル ノークロ アメリカ合衆国 ニューハンプシャー州 03062 ナシユ
ス ア ブルック ヴイレツジ ロード 210 - 8

手続補正書(方式)

3.1.17

平成 年 月 日

特許庁長官 樋 松 敏 殿



1. 事件の表示 平成2年特許願第203805号

2. 発明の名称 データ処理装置における目標指定
リセット法

3. 補正をする者

事件との関係 出 願 人

名 称 デジタル イクイブメント
コーポレーション

4. 代 理 人

住 所 東京都千代田区丸の内3丁目3番1号
電話(代) 211-8741

氏 名 (5995) 弁理士 中 村



5. 補正命令の日付 平成2年10月30日

6. 補正の対象 願書の特許出願人の権
代理権を証明する
明 細 全 図

7. 補正の内容 別紙のとおり

願書に最初に添付した明細 及び図面
月(内容に変更なし)

